

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра технічної кібернетики

«На правах рукопису»

УДК 004.896

До захисту допущено:

Завідувач кафедри

_____ Ігор ПАРХОМЕЙ

«___» _____ 2020 р.

Магістерська дисертація

на здобуття ступеня магістра

**за освітньо-професійною програмою «Інформаційне забезпечення
робототехнічних систем»**

зі спеціальності 126 «Інформаційні системи та технології»

на тему: «Інтелектуальна система оцінювання ринкової вартості житла»

Виконала:

студентка II курсу, групи ІК-91мп

Велітченко Юлія Вікторівна _____

Керівник:

доцент, д.т.н., доц.,

Корнага Ярослав Ігорович _____

Консультант з нормоконтролю:

доцент, к.т.н., доц.,

Пасько Віктор Петрович _____

Рецензент:

професор кафедри ММСА, д.т.н., проф.,

Мухін Вадим Євгенович _____

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів без
відповідних посилань.

Студентка _____

Київ – 2020 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки

Кафедра технічної кібернетики

Рівень вищої освіти – другий (магістерський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційне забезпечення робототехнічних систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Ігор ПАРХОМЕЙ

«__» _____ 2020 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Велітченко Юлії Вікторівні

1. Тема дисертації «Інтелектуальна система оцінювання ринкової вартості житла», науковий керівник дисертації доцент, д.т.н, доцент Корнага Ярослав Ігорович, затверджені наказом по університету від « 26 » жовтня 2020р. № 3132-с

2. Термін подання студентом дисертації _____

3. Об'єкт дослідження – оцінювання ринкової вартості житла

4. Вихідні дані – технічне завдання на розробку інтелектуальної системи оцінювання ринкової вартості житла

5. Перелік завдань, які потрібно розробити – аналіз проблеми оцінювання ринкової вартості житла та існуючих рішень; аналіз і реалізація моделі інтелектуальної системи; розробка програмного забезпечення; дослідження ефективності розробленого методу.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу – шість плакатів

7. Орієнтовний перелік публікацій – дві публікації

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Перевірка на співпадіння	доцент Лісовиченко О.І.		
Нормоконтроль	доцент Пасько В.П.		

9. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Аналіз предметної області		
2	Постановка задачі		
3	Аналіз інформаційного забезпечення		
4	Аналіз алгоритмічного забезпечення		
5	Розробка алгоритмічного забезпечення		
6	Розробка програмного забезпечення		
7	Оформлення пояснювальної записки та додатків		
8	Попередній захист		
9	Нормоконтроль		
10	Перевірка на співпадіння		
11	Захист		

Студент

Юлія Велітченко

Науковий керівник

Ярослав Корнага

АНОТАЦІЯ

У роботі розглянуто проблему в області автоматизованого оцінювання ринкової вартості житла, показано основні особливості існуючих рішень та додатків, визначено їх переваги та недоліки.

Розроблено інтелектуальну систему оцінювання ринкової вартості житла, що базується на нейронній мережі, здатній приймати на вхід кілька типів даних одночасно, а саме: числові, категоріальні та дані зображення. У роботі досліджено, як використовувати методи глибокого навчання для вилучення особливостей зображення, з метою використання цих витягнутих функцій у передбачувальному алгоритмі. Описано структуру мережі та проведено навчання та тестування створеної моделі. Для взаємодії з кінцевим користувачем розроблено веб-додаток. Дана система може бути використана для автоматизації процесу оцінювання вартості житла.

Ключові слова: машинне навчання, нейронна мережа, інтелектуальна система, оцінювання ринкової вартості житла.

Розмір пояснювальної записки – 81 аркуш, містить 40 ілюстрацій, 25 таблиць, 6 додатків.

ABSTRACT

The paper examines the problem in the field of automated assessment of the market value of housing, shows the main features of existing solutions and applications, their advantages and disadvantages.

An intelligent system for the real estate price prediction was designed and built. It is based on a neural network, capable of processing several inputs with different data types simultaneously: numerical, categorical and image data. The paper investigates how to use deep learning methods to extract image features, in order to use these data in the predictive algorithm. The architecture of the network is described, the model was trained and tested. A web application has been developed to interact with the end-user. This system can be used to automate the process of predicting the real estate price.

Keywords: machine learning, neural network, intelligent system, assessment of the real estate market value.

Explanatory note size – 81 pages, contains 40 illustrations, 25 tables, 6 applications.

**Пояснювальна записка
до магістерської дисертації**

на тему: *Інтелектуальна система оцінювання ринкової вартості
житла*

Київ – 2020 року

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	9
ВСТУП.....	10
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ	12
1.1. Об’єкт та предмет дослідження	12
1.2. Огляд існуючих рішень.....	13
1.2.1 Онлайн-калькулятор “DOM.RIA”	13
1.2.2 “Zestimate”	14
1.2.3 “RE/MAX”	16
1.2.4 “Redfin”	17
1.3. Постановка задачі.....	18
Висновки до розділу.....	20
РОЗДІЛ 2. АНАЛІЗ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ	21
2.1. Задача оцінювання ринкової вартості житла.....	21
2.2. Загальний підхід до побудови системи	22
2.3. Штучна нейронна мережа.....	23
2.4. Огляд технологій для розробки системи.....	28
2.4.1 Мова програмування Python	28
2.4.2 Бібліотека машинного навчання Keras	30
2.4.3 Бібліотека машинного навчання TensorFlow	32
2.4.4 Бібліотека машинного навчання Scikit-learn.....	33
2.4.5 Бібліотека машинного навчання Pandas	34
2.4.6 Мова програмування JavaScript та фреймворк Angular.....	35
2.4.7 Бібліотека UI компонентів Angular Material	37
2.4.8 Платформа Node.JS.....	38
Висновки до розділу.....	40

РОЗДІЛ 3. РОЗРОБКА АЛГОРИТМІЧНОГО І ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	41
3.1. Опис інтелектуальної системи	41
3.2. Опис набору даних	43
3.3. Структура проекту та завантаження даних	45
3.4. Проектування мережі для обробки числових та категоріальних даних ..	49
3.5. Проектування згорткової мережі	50
3.6. Побудова фінальної моделі	52
3.7. Опис функціональності веб-додатку	55
Висновки до розділу.....	57
РОЗДІЛ 4. МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП-ПРОЕКТУ	59
4.1. Опис ідеї проекту.....	60
4.2. Технологічний аудит ідеї проекту	62
4.3. Аналіз ринкових можливостей запуску стартап-проекту	62
4.4. Розроблення ринкової стратегії проекту.....	69
4.5. Розроблення маркетингової програми стартап-проекту	71
Висновки до розділу.....	74
ВИСНОВКИ.....	76
ПЕРЕЛІК ПОСИЛАНЬ	78
ДОДАТКИ.....	79
ДОДАТОК А.....	80
ДОДАТОК Б	81

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ШНМ – штучна нейронна мережа

API – application programming interface

AVM – automated valuation model

CNN – convolutional neural networks

CPU – central processing unit

GPU – graphics proccesing unit

ILSVRC – ImageNet large scale visual recognition competition

JS – JavaScript

MLS – multiple listing service

ReLU – rectified linear unit

RMSE – root mean square error

TPU – tensor processing unit

VOC –visual object classes

ВСТУП

Прогнозування ціни товару на ринку може бути складним завданням. Існує багато факторів, які можуть вплинути на ціну будь-якого предмета, який розглядається. Індустрія нерухомості надає ключовий вплив майже на всі аспекти соціальної економіки, враховуючи її великий фінансовий потенціал та тривалий промисловий ланцюг, що стоїть вище та нижче. Точна оцінка вартості будинку – важливе завдання. Для кредитних установ, процес оцінки є критичним для прийняття рішень про позику. Ці установи повинні впевнитись, що вартість майна, яке пропонується під заставу, була пропорційною сумі позики, щоб вони не втратили гроші на дефолтних позиках. Кредитори зазвичай замовляють перевірку та оцінку в процесі кредитування. Оцінка ціни на житло відіграє вирішальну роль у прийнятті рішень щодо нерухомості. Покупці житла та продавці використовують оціночну вартість для оцінки справедливості ціни, що вимагається, та позикодавці використовують ці оцінки для прорахування ризиків.

Тому прогнозування цін на житло розглядається як нова тема останніх десятиліть. Гедонічна регресія є основним методом у даній галузі. «Гедоністична модель» для визначення ринкової вартості житла походить від економічної теорії, висунутої Ланкастером у 1966, яка говорить, що кожен будинок – це сукупність характеристик, які купує споживач: заміна або видалення будь-якої характеристики з колекції принципово змінює товар, що купується, і в результаті може змінити будь-яку прогнозну модель ціни. Потенційні покупці житла мають базовий набір потреб, і загальне рішення про придбання частково базується на наборі характеристик житла, що відповідають цим потребам. Як результат, гедонічна цінова модель може використовувати колекцію числових ознак як предикторів, але цього може бути недостатньо, щоб створити ефективну прогнозовану модель вартості будинку.

Більшість існуючих автоматичних систем оцінювання житла покладаються лише на числові дані, як площа та кількість кімнат. Рішення щодо остаточної ціни приймає агент, що відвідує будинок і оцінює його візуально. Дана дисертація має на меті вивчити застосування методів машинного навчання для прогнозування

ринкової вартості житла, зокрема можливість використання візуальної інформації для покращення прогнозів. Роль методів комп'ютерного зору, які витягують знання із зображень зростає у сучасному житті. Дана робота намагається поширити комп'ютерне бачення на область оцінки нерухомості. Адже використання зображень в прогнозі цін на будинки все ще перебуває на початковій стадії розвитку.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ

1.1. Об'єкт та предмет дослідження

Оцінка ринкової вартості житла є досить складним процесом, що потребує залучення людського ресурсу та передбачає володіння точної статистичної картини, що склалася на ринку. Як правило, цей навик є у досвідчених ріелторів, і найлегше розрахувати вартість житла, звернувшись до них. Існують різні способи дізнатися приблизну ринкову ціну нерухомості. Але навіть якщо запросити кількох професійних оцінювачів або досвідчених ріелторів, то кожен з них назве свою цифру. А реальна ціна продажу, швидше за все, не збіжиться з жодною з них. Так влаштовано ціноутворення на ринку нерухомості.

Каталоги нерухомості використовують визначений перелік числових ознак для опису властивостей. Просторові дані, що проявляються як інформація про місцезнаходження є невід'ємною складовою прогнозування ціни на житло, так як той самий будинок, розташований в різних районах, можна продати за дуже різними цінами. Загалом на ціну житла впливають десятки параметрів, в тому числі характеристики будинку, серед головних можна виділити:

- Кількість спалень;
- Кількість ванних кімнат;
- Площа (тобто квадратні метри);
- Місце розташування;
- Кількість поверхів;
- Тип ремонту;
- Стан будинку.

Більшість існуючих автоматичних систем оцінювання вартості нерухомості використовують лише деякі текстові дані, такі як площа та кількість кімнат. Остаточну оцінку надає людина, що відвідує будинок та оцінює його візуально. Саме під час цього етапу агент оцінює ремонт та загальний стан будинку, що має значний вплив на ціну. Використання даних фотографій будинку може значно підвищити точність прогнозування ціни нерухомості. Для цього можна застосувати згорткові нейронні мережі. Комбінуючи дані зображень, числові та категоріальні

дані (місце розташування), можна отримати найточнішу оцінку ринкової вартості житла. Розробка систем машинного навчання, здатних обробляти змішані дані, може бути надзвичайно складною, оскільки кожен тип даних може вимагати окремих етапів попередньої обробки, включаючи масштабування та нормалізацію.

Об’єкт дослідження – оцінювання ринкової вартості житла.

Предмет дослідження – алгоритми за допомогою яких можна оцінити ринкову вартість житла.

1.2. Огляд існуючих рішень.

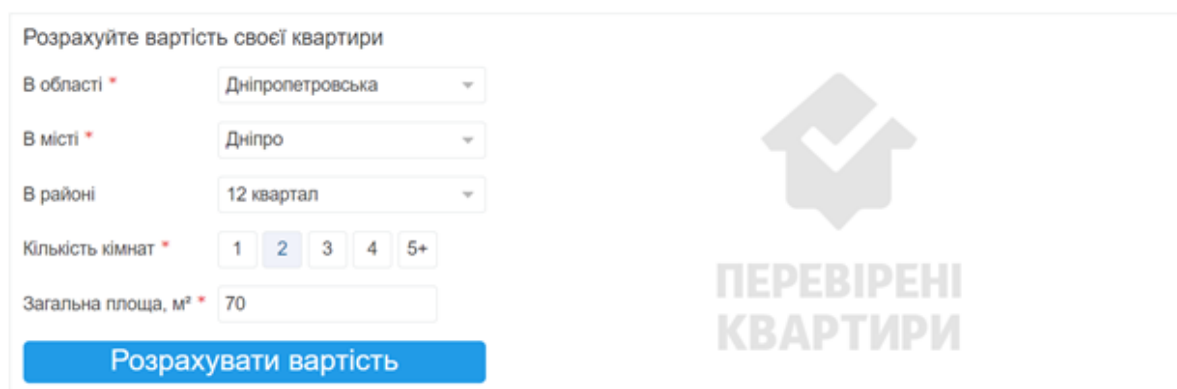
Розглянемо застосунки, що вже створені для оцінювання ринкової вартості житла. Проведемо їх ретельний аналіз та знайдемо головні переваги та недоліки даних систем.

1.2.1 Онлайн-калькулятор “DOM.RIA”

“DOM.RIA” – це український веб-додаток, що є маркетплейсом для продажу та оренди житлових та нежитлових приміщень. Сайт також містить вбудований калькулятор, що дозволяє розрахувати приблизну вартість квартири. Калькулятор є безкоштовним та знаходиться за наступним посиланням:

<https://dom.ria.com/uk/kalkuljator-vartosti-kvarturu/>

Для того, щоб дізнатися вартість квартири, користувачу необхідно ввести кількість кімнат, загальну площу квартири, а також обрати місце розташування, а саме вказати область, місто та район (рис. 1.1).



The image shows a web form titled "Розрахуйте вартість своєї квартири" (Calculate the value of your apartment). The form includes several input fields: "В області" (In the region) with a dropdown menu showing "Дніпропетровська" (Dnipropetrovsk); "В місті" (In the city) with a dropdown menu showing "Дніпро" (Dnipro); "В районі" (In the district) with a dropdown menu showing "12 квартал" (12 district); "Кількість кімнат" (Number of rooms) with buttons for 1, 2, 3, 4, and 5+; and "Загальна площа, м²" (Total area, m²) with a text input field containing "70". A blue button labeled "Розрахувати вартість" (Calculate value) is at the bottom left. To the right of the form is a large, faint watermark logo of a house with a checkmark inside, and the text "ПЕРЕВІРЕНІ КВАРТИРИ" (Verified Apartments).

Рисунок 1.1 Ввід даних в онлайн-калькуляторі “DOM.RIA”

Після цього користувач отримує приблизну вартість квартири (рис. 1.2). Варто звернути увагу, що онлайн калькулятор нерухомості підраховує середньоринкову ціну виходячи з даних для , які вказані в оголошеннях на “DOM.RIA”. Тому онлайн-оцінка квартири не є експертною, а спрямована на орієнтовний результат, оскільки на вартість житла впливає набагато більше факторів.

Розрахуйте вартість своєї квартири

В області *

В місті *

В районі

Кількість кімнат *

Загальна площа, м² *

48 038 \$
(доларів США, станом на жовтень 2020)
*Виконано в якості консультаційної послуги на основі аналізу 3 подібних квартир

Рисунок 1.2 Отримання результату в онлайн-калькуляторі “DOM.RIA”

1.2.2 “Zestimate”

Zillow Group – це американський, один з найвідоміших сайтів торгівлі нерухомістю, що розпочав свою роботу 2006 року. Він простий у використанні, має найбільшу базу даних та пропонує широкий спектр інструментів для власників будинків, включаючи тих, хто планує продати своє житло найближчим часом. Веб-сайт Zillow представляє користувачам “Zestimate”, який представляє послуги оцінки ринкової вартості будинку, виставленого на продаж. Крім того, на сайті перелічено поточні оцінки вартості всіх будинків у їх базі даних, що налічує 97,5 мільйонів будинків по всій країні. “Zestimate” доступний за посиланням:

<https://www.zillow.com/how-much-is-my-home-worth/>

Zillow повідомляє про загальний середній коефіцієнт помилок 4,3%, але він може змінюватися залежно від ринку та кількості точок даних на кожному ринку. Як правило, чим більша кількість будинків на Zillow в на кожному ринку, точнішими стають оцінки.

Після введення адреси можна побачити такі деталі, як кількість проданих подібних будинків у цьому районі, асортимент “Zestimate”, будь-які зміни вартості

за останні 30 днів та прогнози вартості вашого будинку протягом наступного року. Вся ця інформація знаходиться на одній сторінці результатів (рис. 1.3).

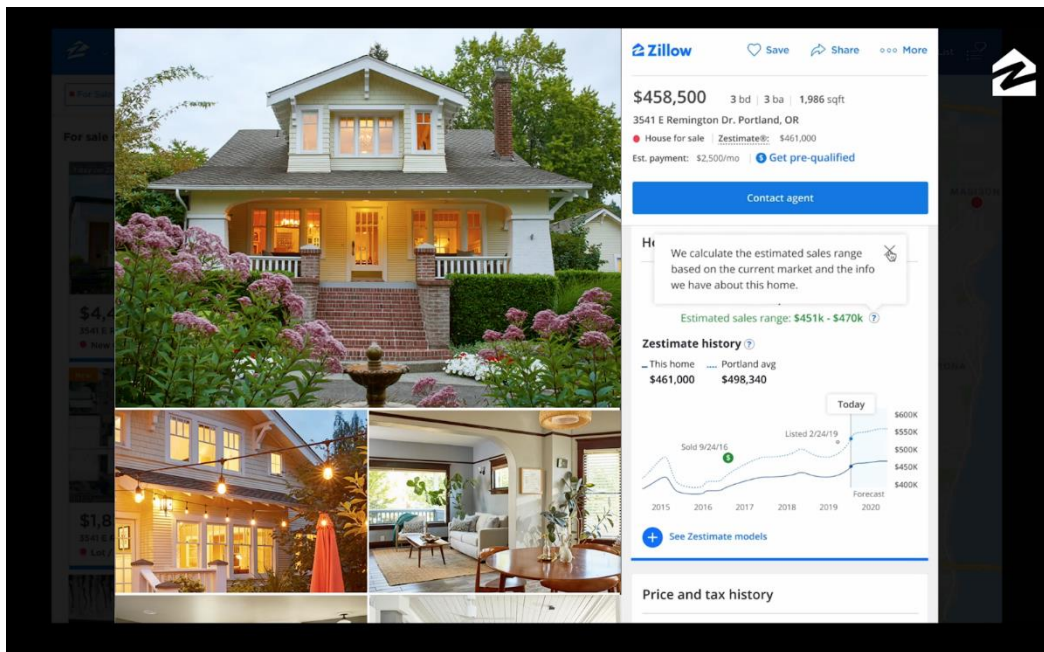


Рисунок 1.3 Сторінка результатів у додатку “Zestimate”

Нещодавно Zillow Group запровадила систему комп'ютерного зору, яка додає точності оцінці від “Zestimate”. Використовуючи обчислювальну потужність від AWS (Amazon Web Services), алгоритми Zillow шукають мільйони доступних зображень житла для ідентифікації предметів, таких як гранітні стільниці, які додадуть вартості будинку. Щоб розрахувати вартість “Zestimate”, Zillow використовує складний і запатентований алгоритм, який включає дані з обліків округів, податкових комітетів та прямі канали сотень різноманітних служб лістингу та посередницьких послуг. “Zestimate” також включає факти та особливості будинку, які власники будинків можуть оновлювати. “Zestimate” враховує такі змінні, як:

- Характеристики будинку, включаючи квадратні метри, розташування або кількість ванних кімнат;
- Унікальні особливості, такі як паркетна підлога, гранітні стільниці або озеленений задній двір;
- Дані на ринку, такі як ціна в лістингу, опис, порівняння будинків у районі та на ринку;

– Позаринкові дані – податкові оцінки, попередні продажі та інші загальнодоступні записи.

Недоліком даного додатку є те, що він використовує датасет будинків США та може бути використаний лише для оцінки житла у даній країні.

1.2.3 “RE/MAX”

Франшиза нерухомості “RE/MAX”, яка охоплює 120 000 ріелторів, на своєму сайті представляє можливість оцінювання вартості житла. “RE/MAX” дозволяє отримати оцінки з трьох веб-сайтів: Eappraisal.com, SmartZip та Zillow. Це економить час користувача від перегляду кількох сайтів для того, щоб отримати оцінки. Сервіс дуже простий у використанні і доступний за наступним посиланням: <https://www.remax.com/home-value-estimates>

Для того, щоб отримати орієнтовну вартість житла, користувачу необхідно ввести адресу будинку, потім вказати кількість спалень, ванних кімнат та загальну площу будинку (рис. 1.4).

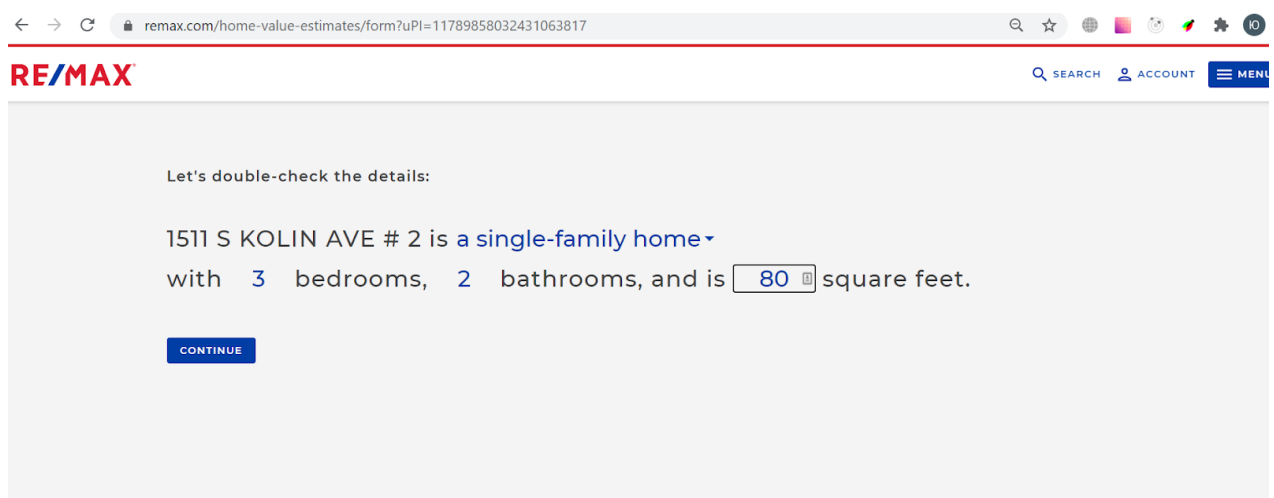


Рисунок 1.4 Ввід даних в додатку “RE/MAX”

“RE/MAX” надсилає дані до трьох сервісів та повертає їх результат (рис. 1.5). Інструмент “RE/MAX” також показує усі будинки, що продаються у даному районі зі схожими характеристиками. З точки зору точності, ці оцінки використовують AVM (автоматизовану модель оцінки) для відображення справжньої вартості будинку. Вони базуються на автоматизованих обчисленнях із загальнодоступних записів, і “RE/MAX” стверджує, що ці цифри мають велику похибку, хоча фактичної кількості не вказано. Наприклад, дані Eappraisal.com можуть бути

неточними, якщо ваш будинок перебуває в стані нерозголошення – дані про продажі не повинні бути в загальнодоступному реєстрі.

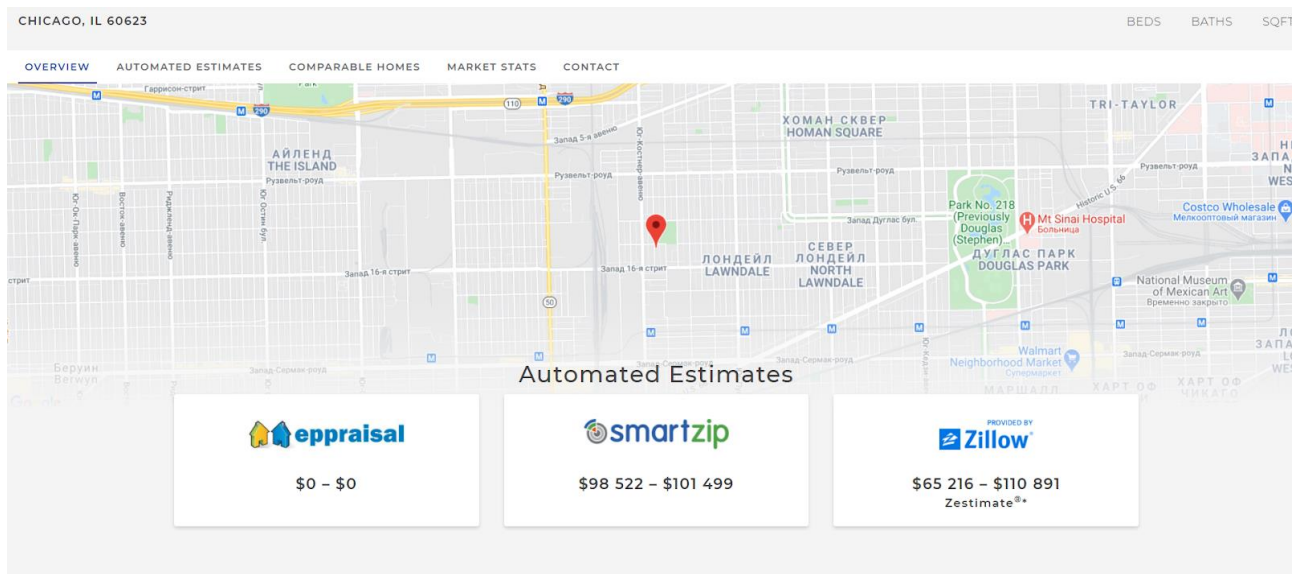


Рисунок 1.5 Отримання результату в додатку “RE/MAX”

Незважаючи на те, що “RE/MAX” оновлює свої списки в режимі реального часу, дані для оцінок можуть відрізнятися залежно від того, з якого веб-сайту вони надходять – наприклад, Eappraisal.com чітко не вказує, як часто оновлює свою інформацію.

Головними недоліками є те, що даний сервіс може бути використаний лише для оцінювання нерухомості в США, а також те, що він враховує лише числові (кількість кімнат, площа) та категоріальні дані(адреса) для оцінки.

1.2.4 “Redfin”

“Redfin” – це посередницька агенція з нерухомості, яка пропонує покупцям та продавцям житла доступ до ріелторів зі знижками з повним спектром послуг та надає онлайн-інструменти, такі як оцінка вартості житла. “Redfin” доступний за наступним посиланням:

<https://www.redfin.com/what-is-my-home-worth>

“Redfin” є одним з найточніших веб-сайтів для оцінки житла. Це досягається завдяки тому, що вони оновлюють дані протягом дня та мають досить низьку похибку для цін на житло. “Redfin” надає найсвіжіші дані: на його веб-сайті та в додатку використовується MLS, або мультілістинг, система, яка об'єднує бази даних різних агентств нерухомості в єдиний інформаційний простір, що

оновлюються кожні 5 хвилин. Оцінки “Redfin” також є одними з найточніших: поточна середня частота помилок 1,71% для будинків на ринку та 5,76% для неринкових. “Redfin” також пропонує скористатися послугами професійної оцінки вартості будинку, агент з нерухомості Redfin може особисто оцінити ваш будинок та надати вам рекомендації щодо ціноутворення на основі його унікальних характеристик.

Як на веб-сайті, так і в додатку легко орієнтуватися. Потрібно просто ввести свою адресу, і інструмент відображає поточну орієнтовну ціну продажу та приблизні ціни протягом наступних п’яти років (рис. 1.6). Крім того, він надає останні продажі житла в цьому районі та середню вартість нерухомості в сусідніх районах.

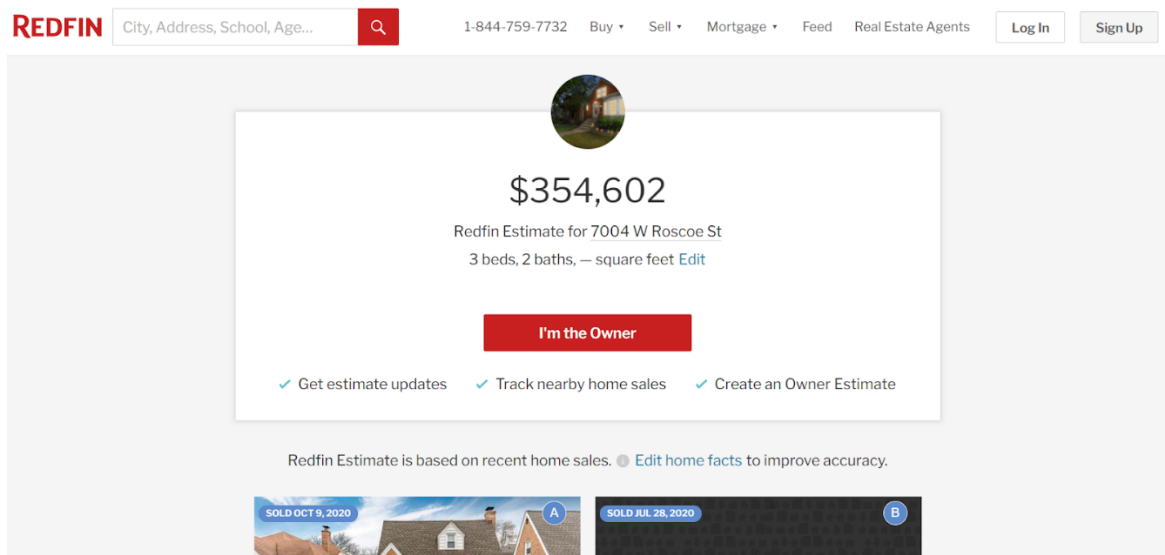


Рисунок 1.6 Сторінка результатів у додатку “Redfin”

Головними недоліками є те, що даний сервіс може бути використаний лише для оцінювання нерухомості в США і приблизна оцінка розраховується, базуючись лише на цінах будинків, що були продані у тому ж районі, тобто жодні інші характеристики не враховуються.

1.3. Постановка задачі

Метою дослідження є підвищення точності оцінювання ринкової вартості житла шляхом використання нейронних мереж. Для цього необхідно спроектувати нейронну мережу та провести її навчання. Для досягнення поставленої мети в магістерській дисертації визначено наступні завдання:

- Провести аналіз систем, які використовують машинне навчання та нейронні мережі;
- Провести додатковий аналіз видів нейромереж;
- Розглянути фактори, що впливають на ринкову вартість житла;
- Сформувати датасет, що містить категоріальні, числові та дані зображень для навчання нейронної мережі;
- Виконати попередню обробку даних;
- Обрати технології та середовище навчання для нейронної мережі;
- Спроекувати архітектуру нейронної мережі, здатної обробляти декілька типів даних;
- Провести навчання нейронної мережі, використовуючи раніше сформований датасет;
- Провести тестування нейронної мережі на тестовому наборі даних;
- Розробити зручний та інтуїтивно-зрозумілий інтерфейс веб-додатку для взаємодії з кінцевим користувачем;
- Обрати технології для реалізації веб-додатку;
- Розробити веб-додаток для оцінювання ринкової вартості житла.

Створений продукт повинен відповідати наступним вимогам:

- Точність оцінки повинна становити більше 80%;
- Система повинна повертати орієнтовану ціну ринкової вартості житла на українському ринку;
- Для визначення ціни система повинна враховувати зовнішній та внутрішній стан будинку;
- Забезпечення швидкого доступу до системи;
- Взаємодія з користувачем повинна відбуватися через інтерфейс українською мовою;
- Робота додатку з форматами зображень: JPEG, PNG;
- Інтуїтивно-зрозумілий інтерфейс веб-додатку для взаємодії з кінцевим користувачем;
- Простота у використанні.

Висновки до розділу

У даному розділі розглянуто проблему оцінювання ринкової вартості житла. Проведено аналіз вже існуючих систем та визначено їх переваги та недоліки. Головними недоліками існуючих рішень є те, що більшість додатків не можуть бути використані для оцінювання нерухомості в Україні і те, що приблизна оцінка розраховується, базуючись лише на цінах будинків, що були продані у тому ж районі, тобто жодні інші характеристики не враховуються. Хоча ці оцінки часто близькі до фактичних цін продажу, в деяких випадках вони є дуже неточними. Одним із ключових факторів, що впливає на вартість будинку, є його внутрішній та зовнішній вигляд, що не враховується при розрахунку автоматичних оцінок вартості. За результатами проведеного аналізу визначено задачі, що повинні бути вирішені для того, щоб створити інтелектуальну систему оцінювання ринкової вартості житла, а також вимоги, яким повинна відповідати створена система.

РОЗДІЛ 2. АНАЛІЗ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ

2.1. Задача оцінювання ринкової вартості житла

Точна оцінка вартості будинку – важливе завдання. Задача оцінка вартості будинку широко вивчалася з багатьох точок зору. Протягом останнього десятиліття було проведено певну роботу для автоматизації процесу оцінки ціни на нерухомість. Порівнюючи різні методи, виявлено, що попередні підходи можна класифікувати на дві основні категорії: моделі на основі дезагрегації даних та основані на агрегуванні даних. Моделі на основі дезагрегації даних намагаються передбачити ціну будинку враховуючи кожен атрибут окремо, так як це відбувається при гедонічній моделі ціноутворення. Моделі агрегування даних враховують усі атрибути для прогнозування ціни, такими моделями є нейронні мережі та регресійні моделі.

Прикладом моделі дезагрегації даних є гедонічна теорія цін, у якій ціна на нерухомість являється функцією атрибутів житла. Атрибути, пов'язані з нерухомістю, визначають набір неявних цін. Граничні неявні значення атрибутів отримують шляхом диференціації гедонічної функції ціни щодо кожного атрибуту. Проблема цього методу полягає в тому, що він не враховує відмінності між різними властивостями в одному і тому ж географічному районі. Ось чому він вважається дуже недостовірним.

Для моделі агрегації даних нейронна мережа є найпоширенішою моделлю. Бін Хаміс у своєму дослідженні порівняв результати роботи нейронної мережі та багатолінійної регресії (MLR). Нейронна мережа досягла вищого коефіцієнту детермінації та нижчої середньоквадратичної помилки, ніж багатолінійна регресійна модель. Порівнюючи результати гедонічної моделі та нейронної моделі, нейронна мережа перевершує гедонічну модель, досягнувши вищого значення коефіцієнту детермінації на 45,348% та нижчої середньоквадратичної помилки на 48,8441%.

Ще одне дослідження було представлено у роботі Лімсомбунчао. Вчений зібрав набір даних про житло з Крайстчерча, Нова Зеландія, що містить 200 точок даних. Потім він створив модель лінійної регресії та модель нейронної мережі,

використовуючи ці дані, і порівняв результуючі середньоквадратичні помилки (RMSE) та коефіцієнт детермінації на тестувальній вибірці. Робота Лімсомбунчао показала набагато вище значення коефіцієнту детермінації (0,9000) для нейронної мережі, ніж для моделі лінійної регресії (0,6192). Її нейромережа також досягла RMSE у 449 111 (новозеландські долари) порівняно з RMSE 876 216 для моделі лінійної регресії.

Враховуючи результати попередніх досліджень вирішено обрати нейронну мережу як метод вирішення задачі оцінювання ринкової вартості житла.

2.2. Загальний підхід до побудови системи

Будинки рідко продаються за вказаною ціною продажу. Взаємодія між покупцем і власником відіграє велику роль у переговорах про остаточний продаж. Ця взаємодія є унікальною для пари покупця-продавця, а тому результат дуже складно передбачити без певних попередніх знань про обох сторін. Очевидно, що зібрати цю величезну кількість індивідуальних та персональних даних практично неможливо без певного типу інфраструктури моніторингу та збору даних, а тому вплив цього фактору лежить поза межами даного дослідження. З іншого боку, одним із факторів у цих переговорах є те, що доцільність вказаної ціни завжди ставиться під сумнів покупцем. Це призводить до наступного припущення: на бажання покупця придбати будинок впливає естетична якість будинку. Будинок зі свіжим шаром фарби, привабливим ландшафтним дизайном та свіжим зовнішнім виглядом може збільшити бажання покупця придбати нерухомість, а отже, і ціну, яку вони готові заплатити. Власність, яка має вицвілу фарбу, зарослий ландшафт чи загалом напівзруйнований вигляд, може бути не такою привабливою для покупця і продатись дешевше, ніж коштував би будинок, якщо б він був у хорошому стані. Архітектурні особливості, стиль або навіть колір можуть також вплинути на покупця, щоб він побачив цінність власності, що має прямий вплив на суму, яку покупець готовий витратити.

Алгоритми прогнозування, які базуються на описових характеристиках, не беруть до уваги жодної візуальної інформації при прогнозуванні ціни продажу. Вони прогнозують ціну на основі функціональних, просторових або часових

особливостей, і часто повертають занижену чи завищену ціну. Це призводить до наступного припущення: Будинки, що продаються за вищу ціну, ніж прогнозована, роблять це частково, тому що покупець “бачить” більше цінності в будинку. І навпаки, будинки, які продають дешевше, ніж передбачали, роблять це частково тому, що покупець “бачить” меншу цінність будинку. Естетичні особливості певним чином впливають на ціну придбання.

Спробуємо вирішити цю проблему прогнозування ціни на будинок, навчивши модель аналізувати образ будинку та робити прогноз щодо естетичної якості будинку. Будемо очікувати, що естетика буде відігравати лише обмежену роль у процесі оцінки, тому не очікуємо, що точність прогнозування для такого класифікатора буде надзвичайно високою. Проте врахування естетичної якості повинна відіграти певну роль, і тому очікуємо, що класифікатор буде працювати на рівні вищому, ніж модель, що не враховує дані зображень.

2.3. Штучна нейронна мережа

Нейронні мережі відображають поведінку людського мозку, дозволяючи комп'ютерним програмам розпізнавати закономірності та вирішувати загальні проблеми в галузі штучного інтелекту, машинного навчання та глибокого навчання. Їх структура імітує спосіб сигналізації біологічних нейронів. Штучні нейронні мережі складаються з шарів нейронів, і містять вхідний шар, один або кілька прихованих шарів та вихідний шар. Будівельним матеріалом для нейронних мереж є штучні нейрони. Це прості обчислювальні одиниці, які мають зважені вхідні сигнали і виробляють вихідний сигнал за допомогою функції активації (рис. 2.1).

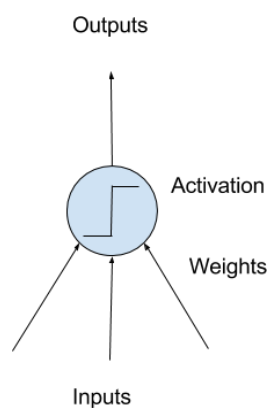


Рисунок 2.1 Модель простого нейрону

Ряд нейронів називається шаром, і одна мережа може мати кілька шарів:

- Вхідний або видимий шар – нижній шар, який приймає вхідні дані з набору даних, називається видимим шаром, оскільки це відкрита частина мережі;
- Прихований шар;
- Вихідний шар – відповідає за виведення значення або вектора значень.

Вибір функції активації на вихідному рівні сильно обмежений типом проблеми, що вирішується. Наприклад:

- Проблема регресії може мати один вихідний нейрон, і нейрон може не мати функції активації;
- Бінарна проблема класифікації може мати один вихідний нейрон і використовувати функцію активації сигмоїдної форми для виведення значення від 0 до 1 для представлення ймовірності прогнозування значення для класу 1. Це може бути перетворено в чітке значення класу за допомогою порогового значення 0,5 і значення прив'язки менше порогового значення до 0, інакше до 1;
- Багатокласова проблема класифікації може мати кілька нейронів у вихідному шарі, по одному для кожного класу.

Розглянемо завдання прогнозування ціни на житло. Як відомо, класифікація використовується для прогнозування приналежності до класу. Для нашої задачі можна визначити наступні класи: дуже дешевий, дешевий, доступний, дорогий, дуже дорогий. Однак дані категорії представляють потенційний діапазон цін на будинок, але не відображають фактичну вартість будинку (рис. 2.2). Для того, щоб передбачити фактичну вартість будинку, необхідно виконати регресію.

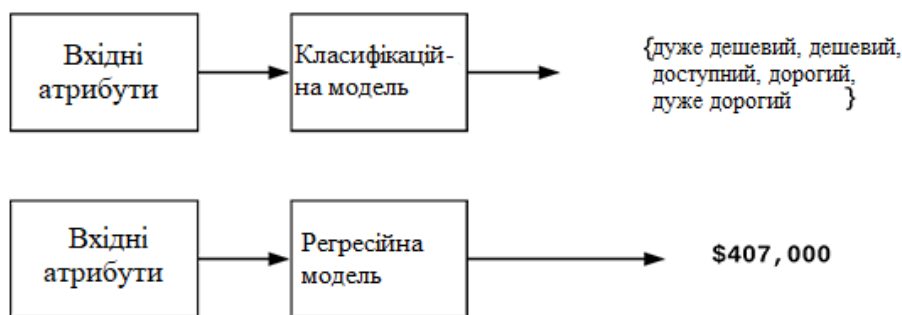


Рисунок 2.2 Візуалізація виконання класифікації і регресії

Для вирішення задачі класифікації та регресії можна використати багатошаровий перцептрон. Багатошаровий перцептрон (MLP) – це перцептрон, який об'єднується з додатковими перцептронами, складеними в кілька шарів, для вирішення складних проблем. Нижче показано MLP з трьома шарами (рис. 2.3). Кожен перцептрон у першому шарі ліворуч (вхідний шар) посилає виходи усім перцептронам у другому шарі (прихований шар), а всі перцептрони у другому шарі відправляють виходи у кінцевий шар праворуч.

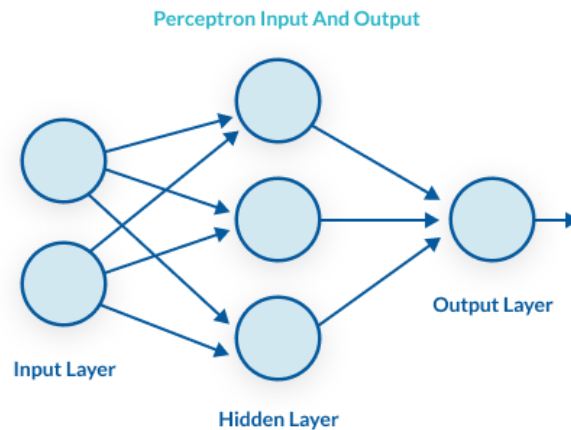


Рисунок 2.3 Багатошаровий перцептрон

Згорткові нейронні мережі (CNN) сьогодні є стандартом для комп'ютерного зору, адже вони постійно встановлюють нові показники ефективності на змаганнях з розпізнавання зображень. Розглянемо декілька архітектур CNN, а також те, як вони можуть бути використані для прогнозуючих моделей у інших сферах шляхом трансферного навчання.

Існує декілька моделей, що можуть бути використані для вирішення поставленої задачі даного дипломного проекту. Одним з перших важливих досягнень в області комп'ютерного зору є AlexNet. AlexNet – це згорткова архітектура нейронної мережі, яка була розроблена для розпізнавання зображень і була однією з перших, яка використовувала техніку дропаут, щоб зменшити перенавчання. AlexNet представляє “глибоку” архітектуру і містить понад 650 000 нейронів, що дає моделі, що використовує цю архітектуру, велику здатність до навчання.

На момент свого заснування у 2012 році AlexNet досягла значних успіхів у точності класифікації зображень ImageNet порівняно з іншими учасниками змагань

з класифікації зображень. Набір даних ImageNet – це добре відомий доступний набір даних, що містить зображення в межах категорій та пов’язані ярлики, які можна використовувати в машинному навчанні та навчанні комп’ютерного зору.

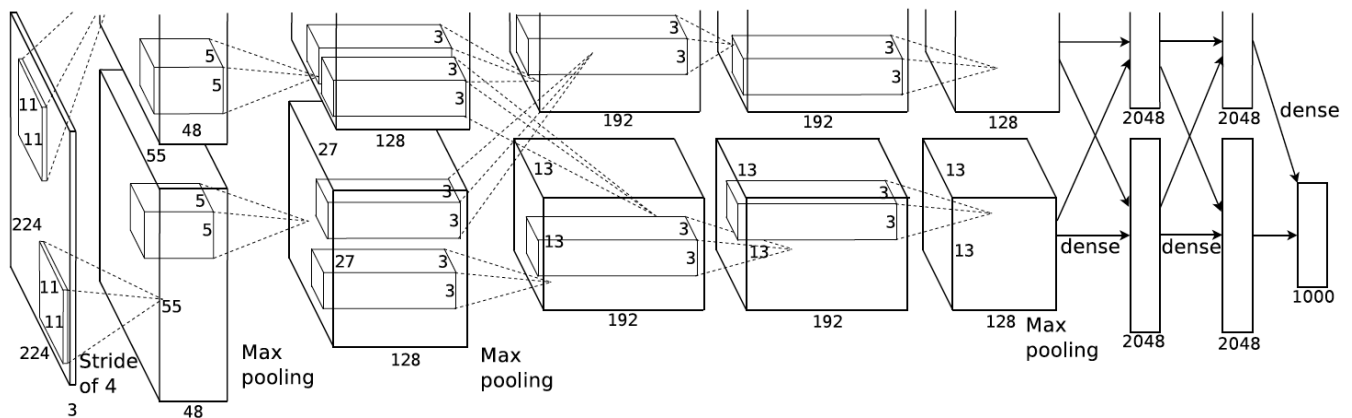


Рисунок 2.4 Архітектура AlexNet

Ще одним значним досягненням у згортковій архітектурі стала робота дослідників групи візуальної графіки Оксфордського університету. Їх архітектура, відома зараз як VGG Net (рис. 2.5), дуже добре продемонструвала себе у змаганні ImageNet 2014, зайнявши 2 місце в класифікації зображень. VGG Net також добре проявляє себе на інших наборах даних. Ця нова архітектура відрізняється від AlexNet використанням менших згорткових фільтрів, які дозволяють створювати більше шарів і, отже, глибшу мережу.

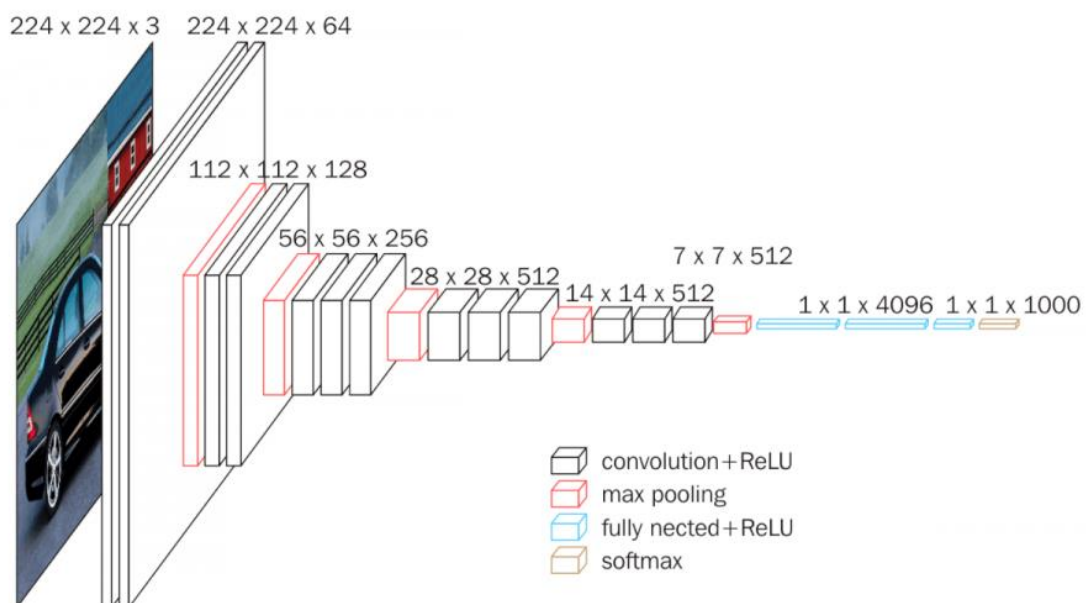


Рисунок 2.5 Архітектура VGG16

Перше місце в ILSVRC за класифікацію зображень у наборі даних ImageNet 2015 року отримала нова архітектура, відома як ResNet. “Res” у ResNet стосується

слова “залишковий”. Побудова цієї моделі була мотивована бажанням побудувати більш глибоку мережу. Глибинні мережі зазвичай складаються з мільйонів параметрів, створених десятками шарів. Через їх розмір важко навчити та управляти цими глибокими мережами. Рішення цієї проблеми вийшло у формі “залишкових функцій”. Ці функції можна навчити і викликати замість декількох шарів у структурі нейронної мережі, що прискорює процес навчання та дозволяє будувати набагато глибшу архітектуру. Виграшна модель ResNet за даними ImageNet 2015 року складалася з 152 шарів, на відміну від згаданої раніше архітектури VGG Net, яка складалася з 41 шару. Архітектура залишкових функцій ResNet дозволяє моделі бути дуже глибокою, але мати керовану кількість компонентів. В одному документі, що узагальнює сучасні досягнення в галузі комп'ютерного зору за останні п'ять років, ResNet згадується як “найсучасніший розвиток архітектур глибинного навчання”.

Трансферне навчання – це процес використання попередньо навченої моделі та застосування цієї моделі до нового датасету. Максим Окваб у своєму дослідженні застосував цю концепцію до набору даних, що складається із зображень з PASCAL VOC, використовуючи архітектуру AlexNet, попередньо навчену на даних ImageNet. Видаливши вихідний шар, вони додали два повністю з'єднаних шари, щоб краще вловити різницю в мітках між двома наборами зображень. Вони зафіксували заздалегідь навчені ваги згорткової бази, а потім “відрегулювали” нові адаптаційні шари до цільового завдання. Відрегульована попередньо навчена модель AlexNet добре працювала з новим набором даних, і автори дійшли висновку, що цей тип мережі може знаходити особливості на зображеннях, які можна виявити при вирішенні подібних завдань і на інших датасетах. У науковому дослідженні класифікації медичних зображень, Шин узагальнив наявні на даний момент варіанти використання архітектур CNN для проблем розпізнавання зображень:

1. Тренування CNN з нуля.
2. Використання моделі “з полиці CNN” або трансферне навчання без зміни моделі CNN.

3. Точне налаштування попередньо тренованої CNN мережі на цільові класи шляхом заморожування певних шарів та імплантації користувацьких шарів у верхній частині мережі.

У магістерській дисертації вирішено застосувати другий варіант, щоб побудувати кращі моделі прогнозування ціни на житло з використанням властивостей, отриманих із зображень за допомогою трансферного навчання.

2.4. Огляд технологій для розробки системи.

2.4.1 Мова програмування Python

Python – інтерпретована, об’єктно-орієнтована, високорівнева мова програмування з динамічною семантикою. Переваги, які роблять Python найбільш підходящим для машинного навчання та проектів на основі ШІ, включають простоту та послідовність, доступ до великої кількості бібліотек та фреймворків для ШІ та машинного навчання, гнучкість, незалежність від платформи та широке співтовариство. Це додає загальної популярності мови. Розглянемо кожен з них більш детально:

- Простота: Python пропонує стислий і читабельний код. Хоча складні алгоритми та універсальні робочі процеси стоять за машинним навчанням та AI, простота Python дозволяє розробникам писати надійні системи. Замість того, щоб зосереджуватись на технічних нюансах мови, розробники докладають свої зусилля для вирішення проблеми ML, замість того щоб зосередитись на технічних нюансах мови. Крім того, Python приваблює багатьох розробників, оскільки його легко вивчити. Код Python є зрозумілим для людей, що полегшує побудову моделей машинного навчання. Загальновизнано, що Python підходить для спільної реалізації, коли залучено кілька розробників. Оскільки Python є мовою загального призначення, він може виконувати набір складних завдань машинного навчання та дозволяє швидко створювати прототипи, що дозволяють протестувати продукт для цілей машинного навчання;

- Широкий вибір бібліотек та фреймворків: впровадження алгоритмів AI та ML може бути складним і вимагає досить багато часу. Щоб скоротити час розробки, програмісти звертаються до ряду фреймворків і бібліотек Python. Python, з його багатим набором технологій, має великий набір бібліотек для штучного інтелекту та машинного навчання. У таблиці 2.1 наведені деякі з них;

Таблиця 2.1

Популярні бібліотеки Python

Випадок використання	Бібліотека
Аналіз та візуалізація даних	NUMPY, SCIPY, PANDAS, SEABORN
Машинне навчання	TensorFlow, Keras, Scikit-learn
Комп'ютерний зір	OpenCV
Обробка природної мови	NLTK, spaCy

- Платформонезалежність: платформонезалежність дозволяє розробникам реалізовувати речі на одній машині та використовувати їх на іншій машині без будь-яких (або лише з мінімальними) змін. Одним із ключових факторів популярності Python є те, що це незалежна від платформи мова. Python підтримується багатьма платформами, включаючи Linux, Windows та macOS. Код Python може бути використаний для створення самостійних виконуваних програм для більшості поширених операційних систем, а це означає, що програмне забезпечення Python можна легко розповсюджувати та використовувати в цих операційних системах без інтерпретатора Python. Більш того розробники зазвичай використовують такі сервіси, як Google або Amazon, для своїх обчислювальних потреб. Однак часто можна зустріти компанії та науковців, які використовують власні машини з потужними графічними процесорами (GPU) для навчання своїх моделей ML. І той факт, що Python не залежить від платформи, робить це навчання набагато дешевшим та простішим;
- Велика спільнота та популярність: інтернет-сховища містять понад 140 000 програмних пакетів Python. Наукові пакети Python, такі як Numpy, Scipy та Matplotlib допомагають розробникам виявляти закономірності у великих

наборах даних. Python настільки надійний, що Google використовує його для сканування веб-сторінок, Pixar – для створення фільмів, а Spotify – для рекомендування пісень. Загальновідомий факт, що спільнота штучного інтелекту Python зростає по всьому світу. Існує активний обмін досвідом, пов'язаний із рішеннями машинного навчання. Для будь-якого завдання, яке у вас може виникнути, досить висока ймовірність того, що хтось інший мав справу з тією ж проблемою.

2.4.2 Бібліотека машинного навчання Keras

Keras – це бібліотека неймереж з відкритим кодом, написана на Python, яка працює поверх Theano або Tensorflow. Він розроблений, щоб бути модульним, швидким та простим у використанні. Його розробив Франсуа Шолле, інженер Google. Keras – це обгортка API високого рівня для API низького рівня, здатна працювати поверх TensorFlow, CNTK або Theano. Keras High-Level API обробляє спосіб створення моделей, визначення шарів або налаштування декількох моделей вводу-виводу. Keras також компілює модель з функціями втрат та оптимізатора, Keras не обробляє API низького рівня, наприклад, створення обчислювального графіку, створення тензорів або інших змінних оскільки це обробляється механізмом “сервера”. На відміну від цього, TensorFlow Core API вимагає роботи з обчислювальними графіками, тензорами, операціями та сесіями, деякі з яких важко зрозуміти, коли ви тільки починаєте працювати з TensorFlow. Є деякі переваги використання низькорівневого TensorFlow Core API, переважно під час налагодження, але, на щастя, можна поєднувати високорівневий і низькорівневий API TensorFlow за потреби.

Keras був створений для зручності користування, модульної роботи, простоти розширення та роботи з Python. API був “розроблений для людей, а не для машин”, і “відповідає найкращим практикам зменшення когнітивного навантаження”. Нейронні шари, функції витрат, оптимізатори, схеми ініціалізації, функції активації та схеми регуляризації – все це окремі модулі, які можна об'єднати для створення нових моделей. Модель є основною структурою даних Keras. У Keras доступні два основних типи моделей: послідовна модель

`Sequential` і клас `Model`, що використовується з функціональним API. Послідовна модель – це лінійний набір шарів, і шари можна описати дуже просто. Розглянемо приклад із документації Keras, який використовує `model.add()` для визначення двох шарів у послідовній моделі (рис. 2.6).

```
import keras
from keras.models import Sequential
from keras.layers import Dense

#Create Sequential model with Dense Layers, using the add method
model = Sequential()

#Dense implements the operation:
#     output = activation(dot(input, kernel) + bias)
#Units are the dimensionality of the output space for the layer,
#     which equals the number of hidden units
#Activation and Loss functions may be specified by strings or classes
model.add(Dense(units=64, activation='relu', input_dim=100))
model.add(Dense(units=10, activation='softmax'))
```

Рисунок 2.6 Створення мережі з використанням послідовної моделі

Кожне визначення шару вимагає лише одного рядка коду. Послідовна модель Keras проста, але обмежена за топологією моделі.

Функціональний API Keras корисний для створення складних моделей, таких як моделі з декількома входами / багатьма виходами, орієнтовані ациклічні графи та моделі зі спільними шарами. Функціональний API використовує ті ж шари, що і послідовна модель, але забезпечує більшу гнучкість їх складання. У функціональному API спочатку визначають шари, а потім створюють модель (рис. 2.7).

```
from keras.layers import Input, Dense
from keras.models import Model

# This returns a tensor
inputs = Input(shape=(784,))

# a layer instance is callable on a tensor, and returns a tensor
x = Dense(64, activation='relu')(inputs)
x = Dense(64, activation='relu')(x)
predictions = Dense(10, activation='softmax')(x)

# This creates a model that includes
# the Input Layer and three Dense Layers
model = Model(inputs=inputs, outputs=predictions)
model.compile(optimizer='rmsprop',
              loss='categorical_crossentropy',
              metrics=['accuracy'])
model.fit(data, labels) # starts training
```

Рисунок 2.7 Створення мережі з використанням функціонального API

Причини використання Keras впливають з його керівних принципів, в першу чергу тих, що стосуються зручності користування. Крім простоти навчання та простоти побудови моделей, Keras пропонує переваги широкого впровадження,

підтримку широкого спектру варіантів розгортання, інтеграцію щонайменше з п'ятьма внутрішніми движками (TensorFlow, CNTK, Theano, MXNet, PlaidML), і потужну підтримку декількох графічних процесорів та розподіленого навчання.

2.4.3 Бібліотека машинного навчання TensorFlow

Створений командою Google Brain, TensorFlow – це бібліотека з відкритим кодом для чисельних обчислень та машинного навчання. TensorFlow поєднує в собі безліч моделей машинного та глибокого навчання, алгоритмів. Він використовує Python, щоб забезпечити зручний інтерфейс API для побудови додатків із фреймворком, в той час виконуючи ці програми у високопродуктивному C ++. Архітектура TensorFlow дозволяє надійно розгортати обчислення на різноманітних платформах (центральної процесорах, графічних процесорах, тензорних процесорах Google), а також від робочих столів до кластерів серверів до мобільних та крайніх пристроїв.

TensorFlow дозволяє створювати графи потоків даних – структури, що описують, як дані рухаються через граф або серію вузлів обробки. Кожен вузол на графі представляє математичну операцію, а кожне з'єднання або ребро між вузлами є багатовимірним масивом даних або тензором. TensorFlow забезпечує все це для програміста за допомогою мови Python. Вузли та тензори в TensorFlow – це об'єкти Python, а програми TensorFlow – програми Python. Однак фактичні математичні операції в Python не виконуються. Python просто спрямовує трафік між фрагментами та забезпечує абстракції програмування високого рівня, щоб з'єднати їх.

Найбільш значною перевагою, що представляє TensorFlow для вирішення задач машинного навчання є абстракція. Замість того, щоб реалізовувати дрібні деталі алгоритмів або з'ясовувати правильні способи приєднання виходу однієї функції до входу іншої, розробник може зосередитись на загальній логіці програми. TensorFlow піклується про деталі за кадром. TensorFlow пропонує додаткові зручні функції для розробників, яким потрібно налагодити та проаналізувати програми TensorFlow. Режим “eager execution” дозволяє оцінювати та модифікувати кожен операцію графа окремо, замість того, щоб будувати весь граф як єдиний

непрозорий об'єкт і оцінювати його роботу загалом. Набір візуалізації TensorBoard дозволяє перевіряти та формувати графіки роботи графів за допомогою інтерактивної веб-панелі інструментів.

Високорівневий API Keras виявився настільки успішним у практиків глибокого навчання, що найновіша версія TensorFlow інтегрує його за замовчуванням. Функціональний API Keras підтримує різноманітні топології з різними комбінаціями входів, виходів та шарів. Ще одною перевагою TensorFlow є те, що він сумісний з багатьма мовами програмування, такими як C ++, JavaScript, Python, C #, Ruby та Swift. Це дозволяє користувачеві працювати в середовищі, в якому йому комфортно.

2.4.4 Бібліотека машинного навчання Scikit-learn

Sklearn або Scikit-learn – одна з найкорисніших доступних бібліотек із відкритим кодом для машинного навчання в Python. Бібліотека Scikit-learn – це колекція найефективніших інструментів статистичного моделювання та машинного навчання. Деякі з цих інструментів включають регресію, класифікацію, зменшення розмірності та кластеризацію. Бібліотека використовує уніфікований та послідовний інтерфейс Python для реалізації різних алгоритмів попередньої обробки, машинного навчання, візуалізації та перехресної перевірки. Sklearn був розроблений Девідом Курно в 2007 році в рамках проекту Google Summer Code. Scikit-learn – це проект, де кожен може внести свій внесок у його розвиток. Microsoft, Intel та NVIDIA є одними з головних спонсорів проекту. Бібліотека машинного навчання Scikit-learn має безліч функцій для спрощення машинного навчання. Розглянемо деякі з них:

- Алгоритми контрольованого навчання: будь-який керований алгоритм машинного навчання має дуже високу можливість належати до бібліотеки Scikit-learn. Інструментарій Scikit-learn має набір таких керованих алгоритмів навчання, що включає узагальнені лінійні моделі, такі як лінійна регресія, дерева рішень, методи опорних векторів та байєсівські методи;

- Алгоритми навчання без нагляду: ця колекція алгоритмів включає факторинг, кластерний аналіз, аналіз основних компонентів та нейронні мережі, що навчаються без учителя;
- Вилучення ознак: використовуючи Scikit-learn, можна вилучати ознаки з тексту та зображень;
- Перехресна перевірка: точність та достовірність контрольованих моделей на невидимих даних можна перевірити за допомогою Scikit-learn;
- Зменшення розмірності: за допомогою цієї функції кількість атрибутів у даних можна зменшити для подальшої візуалізації, узагальнення та вилучення ознак;
- Кластеризація: ця функція дозволяє групувати нерозмічені дані;
- Ансамблеві методи: за допомогою цієї функції можна поєднувати прогнози кількох контрольованих моделей.

2.4.5 Бібліотека машинного навчання Pandas

Pandas – це пакет Python, що надає швидкі, гнучкі та виразні структури даних, покликані зробити роботу зі структурованими (табличними, багатовимірними, потенційно неоднорідними) та даними часових рядів легкими та інтуїтивно зрозумілими. Він має на меті стати фундаментальним будівельним елементом високого рівня для практичного аналізу даних. Pandas добре підходить для багатьох різних типів даних:

- Табличні дані;
- Впорядковані та невпорядковані дані часових рядів;
- Довільні дані матриці з мітками рядків і стовпців;
- Будь-яка інша форма спостережних / статистичних наборів даних.

Дані насправді взагалі не потрібно мітити, щоб розміщувати їх у структурі даних Pandas. Pandas має дві основні структури даних: Series (одномірні) і DataFrame (двомірні), вони підходять для переважної більшості задач.

2.4.6 Мова програмування JavaScript та фреймворк Angular

JavaScript – це динамічна мова комп'ютерного програмування. Він легкий і дозволяє клієнтському додатку взаємодіяти з користувачем та створювати динамічні сторінки. Це інтерпретована мова програмування з об'єктно-орієнтованими можливостями. Розглянемо головні переваги:

- Швидкість: клієнтський JavaScript дуже швидкий, оскільки його можна негайно запустити в клієнтському браузері. Якщо не потрібні зовнішні ресурси, JavaScript не перешкоджає мережевим викликам до сервера;
- Простота: JavaScript порівняно простий у вивченні та реалізації;
- Популярність: JavaScript використовується повсюдно в Інтернеті. Щоб вивчити JavaScript, існує незліченна кількість ресурсів, на які можна звернутися. GitHub та StackOverflow демонструють ескалантну кількість проектів із використанням JavaScript, і очікується, що популярність лише зросте;
- Сумісність: JavaScript чудово поєднується з іншими мовами і може бути використаний у величезній кількості різноманітних додатків;
- Навантаження сервера: будучи на стороні клієнта, зменшується навантаження на сервері веб-сайту;
- Дає можливість створювати розширені інтерфейси.

Angular, що підтримується Google, – це інженерна платформа з відкритим кодом, яка використовується для побудови користувацьких інтерфейсів. Він був розроблений інженерами Google у 2009 році. Протягом тривалого часу цей фреймворк розглядався розробниками як інструмент інтерфейсу, який використовується за замовчуванням, завдяки своїй універсальності та можливості поєднувати бізнес-логіку та елементи інтерфейсу. Він залишається однією з найпопулярніших технологій розробки програмного забезпечення, пропонуючи різні функції, які допомагають масштабувати, оптимізувати та пришвидшувати веб-додатки. Розглянемо головні переваги:

- Функціональність, що відразу готова до використання: налаштування Angular за замовчуванням надає велику кількість готового функціоналу.

Сюди входять інструменти для маршрутизації, так що можна легко отримати дані, які ви хочете представити у своєму додатку. Попередньо налаштоване середовище Angular не тільки допомагає в розробці, але й полегшує тестування. Не потрібно використовувати будь-які сторонні бібліотеки для створення базових функцій додатку. Все, що потрібно – це офіційна бібліотека, яку надає команда Angular. Це означає, що можна очікувати кращу безпеку та вищу якість коду;

- TypeScript: Angular побудований за допомогою TypeScript. Головною перевагою цієї мови є те, що вона допомагає розробникам підтримувати свій код чистим та зрозумілим. Помилки легше виявити та усунути завдяки можливості бачити типові помилки під час введення. Це робить його швидшим, коли справа доходить до налагодження, а також це дає змогу підтримувати велику кодову базу – те, що особливо корисно у великих корпоративних проектах;
- Цілісність та послідовність: на відміну від React, Angular – це повноцінний цілісний фреймворк. Ключова особливість полягає в тому, що є єдиний стандарт написання компонентів, сервісів та модулів. На практиці це створює послідовність у всій основі коду, що є важливою метою до якої слід йти, а також це дозволяє уникнути ситуації, коли інші розробники задаються питанням, як слід щось побудувати. Для подальшого підвищення узгодженості команда Angular розробила інструмент CLI, який можна використовувати для створення певних повторюваних блоків коду з командного рядка. Крім того, документація фреймворку служить зручним актуальним довідковим матеріалом для розробників;
- Продуктивність: приємним бонусом послідовності є підвищення продуктивності. Розробникам не потрібно витрачати цінний час, намагаючись з'ясувати, що робить функція, утиліта чи компонент, і як тільки вдасться освоїти, як писати компонент, написати новий достатньо легко, дотримуючись тих самих загальних вказівок та структури коду. Висока читабельність коду Angular також полегшує новим розробникам можливість одразу перейти до роботи на проекті. Типізація дозволяє розробникам раніше

ловити потенційні помилки. Завдяки інтеграції в IDE, такі як VS Code та WebStorm, TypeScript поступово перекомпілює та підкреслює помилки під час написання коду. Кожен із цих факторів сприяє скороченню часу, а отже витрат на розробку;

- Підтримка оновлень: Angular має чудову підтримку оновлення коду кількома способами. По-перше, при переході від однієї основної версії до іншої всі пакети, пов'язані з Angular, оновлюються одночасно – це означає, що включені також HTTP, маршрутизація та Angular Material. Оновлення є простим, і його зазвичай можна досягти, використовуючи лише одну команду “ng update”. Це означає, що ви витрачаєте менше часу і ви відразу знаєте яку версію слід використовувати, або не задаєтеся питанням, коли певний пакет із відкритим кодом буде оновлений до останньої версії;
- Модульна структура розробки: модульність Angular дозволяє розробникам ефективно розділити код на модулі, що дозволяє легко організувати функціональність програми та створити багаторазові фрагменти коду, що насамперед може значно скоротити час і витрати на розробку. Спільнота Angular підтримує такий підхід за допомогою різноманітних готових компонентів. Модулі також дозволяють ефективно розподіляти роботу з розробки між командами, забезпечуючи при цьому упорядкованість та чистоту коду, а також дозволяючи плавно масштабувати програму.

2.4.7 Бібліотека UI компонентів Angular Material

Angular Material – це колекція готових до використання, добре перевірених та протестованих компонентів та модулів інтерфейсу, які відповідають принципам Material Design Google. Він містить ряд компонентів інтерфейсу користувача, таких як шаблони навігації, елементи керування формою, кнопки та індикатори. Компоненти добре адаптовані для різних браузерів, гарно задокументовані та написані на основі останніх рекомендацій. Розроблені спільнотою Angular, ці модулі спрощують робочий процес для команди, дозволяючи розробникам швидко розробляти та додавати нові елементи. Сам Матеріал Дизайн – це адаптивна система вказівок, компонентів та інструментів, що підтримують найкращі практики

проектування інтерфейсу користувача. За підтримки відкритого коду Material впорядковує співпрацю між дизайнерами та розробниками та допомагає командам швидко створювати красиві продукти, що є зручними та інтуїтивно зрозумілим для користувачів. Використовуючи цю бібліотеку можна створювати як швидкі прототипи так і цілісні індивідуальні дизайни стилізовані під бренд чи особливі вимоги додатку.

2.4.8 Платформа Node.JS

JavaScript керував вебом як найпопулярніша мовою програмування на стороні клієнта. Те, що ми можемо робити з JavaScript на стороні сервера, було лише концепцією до введення Node.js у 2009 році. Завдяки перевагам Node.js розробка веб-додатків стала надзвичайно простою, економічною та надзвичайно ефективною під час роботи JavaScript на стороні сервера.

Node.js – це середовище виконання з відкритим кодом JavaScript, побудоване на движку V8 Chrome. Він подійно-орієнтований з неблокуючим вводом-виводом, що робить його легким, ефективним і надзвичайно швидким для розробки веб-додатків. Розглянемо головні переваги:

- Висока продуктивність для додатків реального часу: веб-додатки на базі Node.js демонструють значну продуктивність, адже Node.js має таку перевагу, як багатозадачність. На відміну від інших платформ, його однопотокова подійно-орієнтована архітектура ефективно обробляє кілька одночасних запитів, не засмічуючи оперативну пам'ять. Більше того, його цикли подій та неблокуючі операції вводу-виводу дозволяють виконувати код із швидкістю, яка суттєво впливає на загальну продуктивність програми. Також значної швидкості додає і те, що Node.js побудований на двигуні V8 Google Chrome і написаний на C ++. Він розбиває функції Javascript на машинні коди з високою ефективністю та швидкістю. Google вкладає значні кошти у свій двигун, щоб покращити продуктивність, безпеку та забезпечити чудову підтримку сучасних функцій JavaScript. Це безпосередньо впливає на продуктивність V8 і, зрештою, Node.js;

- Масштабованість: Node.js має безліч корисних функціональних реалізацій, таких як кластерний модуль. Він полегшує балансування навантаження на декількох ядрах процесора. Крім того, Node.js використовує неблокуючий механізм циклу подій, який пропонує високу масштабованість і дозволяє серверу безперешкодно обробляти запити. З точки зору розробки, Node.js дозволяє використовувати мікросервіси, що дозволяє розділити програму на менші частини. Node.js та мікросервіси дозволяють сучасним програмам масштабуватися вгору та вниз за потребою та допомагають компаніям досягти високої продуктивності з меншою кількістю ресурсів;
- Велика спільнота: мільйони розробників роблять свій внесок у спільноту Node.js, а отже можна розраховувати на значну підтримку експертів з усього світу для вирішення навіть найскладніших проблем. Ця активна спільнота підтримується багатьма провідними технологічними гігантами, такими як Amazon, Google, Facebook та Netflix. Вони зробили помітний внесок у вигляді багатьох рішень з відкритим кодом. NPM, менеджер пакетів для JavaScript, є найбільшим реєстром менеджерів пакетів у світі. Він пропонує численні інструменти та бібліотеки;
- Покращує час відгуку додатка та підвищує продуктивність: кількість запитів має значний вплив на час відгуку та продуктивність програми. Node.js зі своєю однопотоковою моделлю циклу подій пропонує неблокуючу асинхронну архітектуру, не створюючи більше потоків та використовуючи менше ресурсів. Це допомагає підвищити респонсивність програми, оскільки вона може працювати з кількома одночасними користувачами;
- Скорочує час завантаження за допомогою швидкого кешування: Node.js дозволяє розробникам зменшити навантаження задачами та повторне виконання коду за допомогою модуля кешування. Отже, кожного разу, коли перший модуль веб-програми отримує запит, він потрапляє в кеш пам'яті в додатку. Таким чином, за частку мікросекунд користувачі отримують швидкий доступ до веб-сторінок, не чекаючи надто довго.

Висновки до розділу

У даному розділі розглянута задача оцінювання ринкової вартості житла. Проаналізовано різні методи її вирішення. Виявлено, що попередні підходи можна класифікувати на дві основні категорії: моделі на основі дезагрегації даних та основані на агрегуванні даних. Визначено загальний підхід до побудови системи, зокрема використання нейронної мережі. Вирішено навчити модель аналізувати образ будинку та враховувати інформацію щодо естетичної якості будинку при прогнозуванні ціни.

Також проведено огляд технологій, що планується використовувати для розробки веб-додатку та нейронної мережі. Визначено переваги кожної з технологій. Для розробки та тренування нейронної мережі вирішено використовувати мову програмування Python, бібліотеки машинного навчання TensorFlow, Keras, Scikit Learn, Pandas. Для розробки веб-додатку вирішено використовувати мову програмування JavaScript, Node.js, фреймворк Angular та бібліотеку UI компонентів Angular Material.

РОЗДІЛ 3. РОЗРОБКА АЛГОРИТМІЧНОГО І ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1. Опис інтелектуальної системи

Набір даних для визначення ціни житла, що використано для навчання мережі, включає не тільки числові та категоріальні дані, але й дані зображень – різні типи даних називають змішаними даними, оскільки модель повинна мати змогу приймати наші численні входи, які мають різний тип і робити обчислювальні прогнози на цих входах (рис. 3.1). Зображення будинків представляють здавалося б невидимий ресурс для прогнозування цін на будинки.

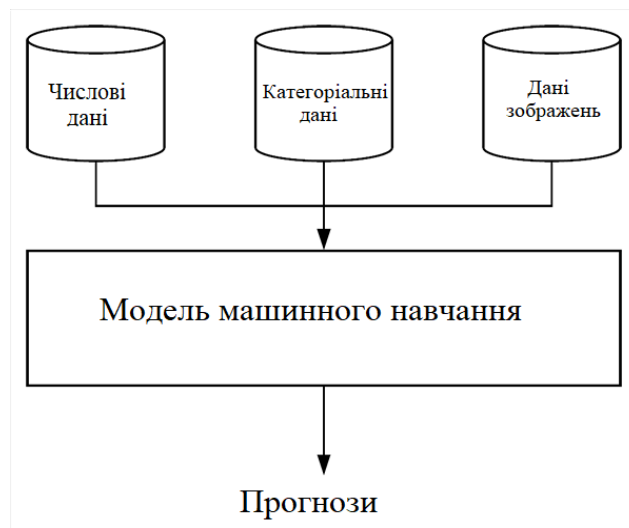


Рисунок 3.1 Багатовхідна модель

У машинному навчанні змішані дані стосуються концепції наявності декількох типів незалежних даних. Наприклад, припустимо, ми – інженери машинного навчання, які працюють у лікарні над розробкою системи, здатної класифікувати стан здоров’я пацієнта. Ми мали б декілька типів вхідних даних для даного пацієнта, включаючи:

- Цифрові значення, такі як вік, частота серцевих скорочень, артеріальний тиск;
- Категорійні значення, включаючи стать та етнічну приналежність;
- Дані зображення, такі як МРТ, рентген тощо.

Усі ці значення складають різні типи даних, однак наша модель машинного навчання повинна мати можливість приймати ці “змішані дані” та робити точні

прогнози щодо них. Розробка систем машинного навчання, здатних обробляти змішані дані, може бути надзвичайно складною, оскільки кожен тип даних може вимагати окремих етапів попередньої обробки, включаючи масштабування, нормалізацію та конструювання ознак. Робота зі змішаними даними все ще залишається відкритою сферою досліджень і часто сильно залежить від конкретного завдання чи кінцевої мети.

Keras здатний обробляти кілька входів (і навіть кілька виходів) за допомогою свого функціонального API. Функціональний API, на відміну від послідовного API, можна використовувати для визначення набагато складніших моделей, які не є послідовними, зокрема:

- Моделі з декількома входами;
- Моделі з декількома виходами;
- Моделі, що є одночасно кількома вхідними і вихідними;
- Направлені ациклічні графи;
- Моделі зі спільними шарами.

На рис. 3.2 зображена модель нейронної мережі для вирішення задачі оцінювання ринкової вартості житла.

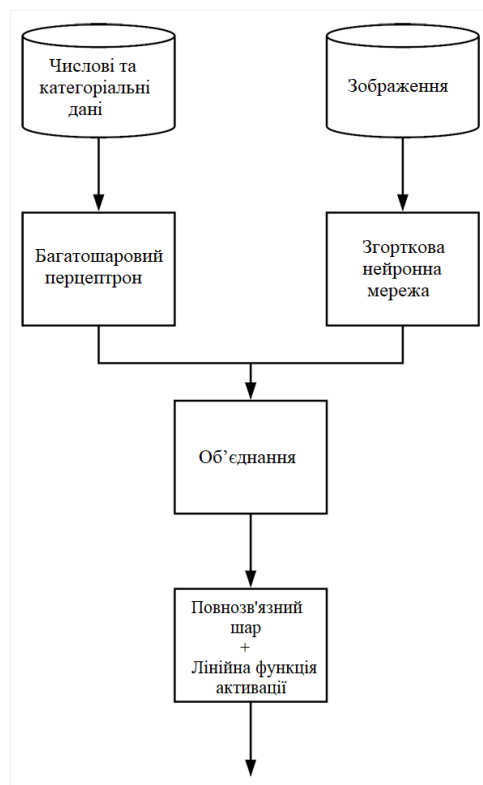


Рисунок 3.2 Модель нейронної мережі

Для побудови мережі з багатьма входами знадобляться дві гілки:

- Першою гілкою є простий багатошаровий перцептрон (MLP), призначений для обробки категоріальних / числових входів;
- Другою гілкою є згорткова нейронна мережа, яка працюватиме над даними зображення;
- Потім ці гілки будуть об'єднуються разом, щоб сформувати остаточну багатовхідну модель.

3.2. Опис набору даних

Моделі глибокого навчання вимагають великої кількості навчальних даних для того, щоб повертати точний результат. Є багато наборів даних про житло, що доступні за попередніми науковими проектами та конкурсами, але жоден з них не має як описових даних, так і зовнішніх зображень відповідних будинків. Зібраний набір даних складається з 2140 зразків зображень будинків, що були виставлені на продаж на території України під час формування датасету. Набір даних включає як числові / категоріальні дані, так і дані зображень для кожного будинків у наборі даних. Для кожного будинку міститься по 4 зображення. Крім того, датасет містить текстовий файл з текстовими метаданими набору даних. Кожен рядок у файлі відповідає номеру будинку в порядку. Числові та категорійні атрибути включають:

- Кількість спалень;
- Кількість ванних кімнат;
- Площа (тобто квадратні метри);
- Поштовий індекс.

Всього для кожного будинку передбачено чотири зображення:

- Спальня (рис. 3.3);
- Ванна кімната (рис. 3.4);
- Кухня (рис. 3.5);
- Фронтальний вид на будинок (рис. 3.6).

Цей набір даних було зібрано вручну із загальнодоступної інформації на веб-сайтах, що продають будинки. Зібрані зображення були різної якості. Багато зображень були такими, як очікувалося, наприклад, гарне представлення будинку з

боку бордюру, щоб зробити його привабливим для потенційного покупця. Однак деякі зображення не були корисними для оцінки естетичної цінності будинку. Багато будинків ще будувалися.



Рисунок 3.3 Зображення спальні з датасету



Рисунок 3.4 Зображення ванної кімнати з датасету

Деякі фотографії були зроблені в портретній орієнтації, тоді як більшість фотографій зроблено в альбомній орієнтації. Через ці фотопорушення знадобилося вручну обрізати будинки подальше від будь-яких сусідніх будинків, видаляти з датасету ті будинки, що мали зображення, з яких не було видно будинку.



Рисунок 3.5 Зображення кухні з датасету



Рисунок 3.6 Зображення фронтального виду на будинок з датасету

Як результат, датасет не містить як повторних, так і відсутніх даних. Ціна будинку в наборі даних коливається від 27 000 до 3 350 000 доларів. У табл. 3.1 наведені деякі статистичні дані про набір даних.

Таблиця 3.1

Статистичні дані про датасет

Показник	Середнє значення	Мінімальне значення	Максимальне значення
Ціна будинку (USD)	230000	27000	3350000
Площа будинку (м2)	212	65	1466
Кількість спалень	3.36	1	9
Роздільна здатність зображення	801×560	250×187	1484×1484

Необхідно створити дві гілки мережі для обробки кожного типу даних, які потім необхідно об'єднати в кінці, щоб отримати остаточний прогноз ціни на будинок. Таким чином, зможемо використовувати Keras для обробки як кількох входів, так і змішаних даних.

3.3. Структура проекту та завантаження даних

На рис. 3.7 представлена структура проекту. Папка набору даних Houses містить набір даних про ціни на будинки. Розглянемо файли папки ruimagesearch:

- datasets.py: обробляє завантаження та попередню обробку числових / категоріальних даних, а також даних зображень;

- `models.py`: містить багаточаровий перцептрон та згорткову нейронну мережу. Ці компоненти є вхідними гілками для нашої багатовхідної, змішаної моделі даних;
- `mixed_training.py`: наш навчальний скрипт, що використовує функції модуля `pyimagesearch` для завантаження, розділення даних та об'єднання двох гілок у нашу мережу, додавання заголовка. Потім він навчатиме та оцінюватиме модель.

```

1.  $ tree --dirsfirst --filelimit 10
2.  .
3.  ├── Houses-dataset
4.  │   ├── Houses\ Dataset [2141 entries]
5.  │   └── README.md
6.  ├── pyimagesearch
7.  │   ├── __init__.py
8.  │   ├── datasets.py
9.  │   └── models.py
10. └── mixed_training.py
11.
12.  3 directories, 5 files

```

Рисунок 3.7 Структура проекту

Завантажимо числові та категоріальні дані (рис. 3.8). Для цього використаємо `pandas`, пакет Python, для зчитування даних з CSV файлу.

```

[>>> import pandas as pd
[>>> cols = ["bedrooms", "bathrooms", "area", "zipcode", "price"]
[>>> inputPath = "HousesInfo.txt"
[>>> df = pd.read_csv(inputPath, sep=" ", header=None, names=cols)
[>>> df.head()
   bedrooms  bathrooms  area  zipcode  price
0         4         4.0  4053   85255  869500.0
1         4         3.0  3343   36372  865200.0
2         3         4.0  3923   85266  889000.0
3         5         5.0  4022   85262  910000.0
4         3         4.0  4116   85266  971226.0
[>>>

```

Рисунок 3.8 Завантаження числових та категоріальних даних

Обробимо попередньо завантажені числові та категоріальні дані. Для цього визначимо функцію `load_house_attributes` (рис. 3.9). Ця функція зчитує числові / категоріальні дані із датасету вигляді CSV-файлу. Дані фільтруються, щоб усунути дисбаланс. Деякі поштові індекси представлені одним чи двома будинками, тому продовжуємо і видаляємо будь-які записи, де для поштового індексу менше 25 будинків. Результат – більш точна модель згодом.


```

10. def load_house_attributes(inputPath):
11.     # initialize the list of column names in the CSV file and then
12.     # load it using Pandas
13.     cols = ["bedrooms", "bathrooms", "area", "zipcode", "price"]
14.     df = pd.read_csv(inputPath, sep=" ", header=None, names=cols)
15.
16.     # determine (1) the unique zip codes and (2) the number of data
17.     # points with each zip code
18.     zipcodes = df["zipcode"].value_counts().keys().tolist()
19.     counts = df["zipcode"].value_counts().tolist()
20.
21.     # loop over each of the unique zip codes and their corresponding
22.     # count
23.     for (zipcode, count) in zip(zipcodes, counts):
24.         # the zip code counts for our housing dataset is *extremely*
25.         # unbalanced (some only having 1 or 2 houses per zip code)
26.         # so let's sanitize our data by removing any houses with less
27.         # than 25 houses per zip code
28.         if count < 25:
29.             idxs = df[df["zipcode"] == zipcode].index
30.             df.drop(idxs, inplace=True)
31.
32.     # return the data frame
33.     return df

```

Рисунок 3.9 Фільтрування даних

Наступним кроком визначимо функцію для попередньої обробки даних (рис. 3.10). Застосуємо мінімально-максимальне масштабування до числових ознак та перетворимо категорійні дані у формат one-hot encoding. Об'єднаємо числові та категорійні дані та повернемо їх.

```

35. def process_house_attributes(df, train, test):
36.     # initialize the column names of the continuous data
37.     continuous = ["bedrooms", "bathrooms", "area"]
38.
39.     # perform min-max scaling each continuous feature column to
40.     # the range [0, 1]
41.     cs = MinMaxScaler()
42.     trainContinuous = cs.fit_transform(train[continuous])
43.     testContinuous = cs.transform(test[continuous])
44.
45.     # one-hot encode the zip code categorical data (by definition of
46.     # one-hot encoding, all output features are now in the range [0, 1])
47.     zipBinarizer = LabelBinarizer().fit(df["zipcode"])
48.     trainCategorical = zipBinarizer.transform(train["zipcode"])
49.     testCategorical = zipBinarizer.transform(test["zipcode"])
50.
51.     # construct our training and testing data points by concatenating
52.     # the categorical features with the continuous features
53.     trainX = np.hstack([trainCategorical, trainContinuous])
54.     testX = np.hstack([testCategorical, testContinuous])
55.
56.     # return the concatenated training and testing data
57.     return (trainX, testX)

```

Рисунок 3.10 Попередня обробка даних

Завантажимо набір даних зображень. На рис. 3.11 представлено приклад зображення, що повинен бути отриманий для кожного з будинків, щоб провести навчання нейронної мережі. Перша гілка моделі приймає одне зображення – монтаж чотирьох зображень з будинку. Використовуючи змонтоване зображення в поєднанні з числовими / категоріальними даними, що є вхідними даними для другої

гілки, модель потім використовує регресію для прогнозування вартості будинку за допомогою фреймворка Keras.

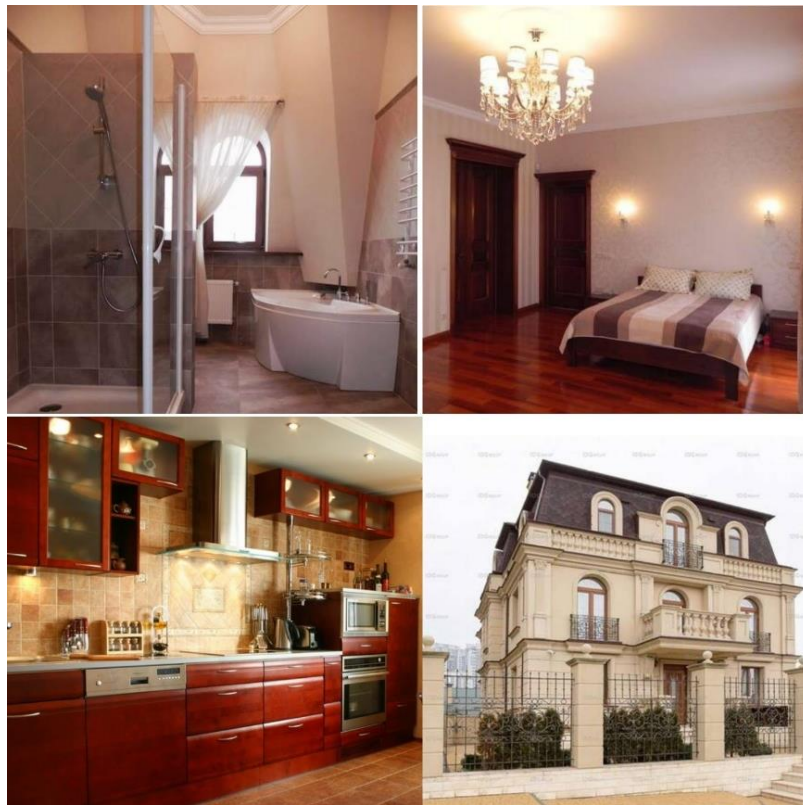


Рисунок 3.11 Приклад зображення для навчання мережі

Визначимо допоміжну функцію для завантаження вхідних зображень (рис. 3.12). Дана функція має три цілі:

- Завантажує всі фотографії з набору даних (4 зображення для кожного будинку);
- Створює одне змонтоване зображення з чотирьох фотографій. Монтаж завжди буде влаштований, як показано на прикладі зображення;
- Додає всі ці змонтовані зображення до списку / масиву та повертає до функції виклику.

```
59. def load_house_images(df, inputPath):
60.     # initialize our images array (i.e., the house images themselves)
61.     images = []
62.
63.     # loop over the indexes of the houses
64.     for i in df.index.values:
65.         # find the four images for the house and sort the file paths,
66.         # ensuring the four are always in the *same order*
67.         basePath = os.path.sep.join([inputPath, "{}_{}".format(i + 1)])
68.         housePaths = sorted(list(glob.glob(basePath)))
```

Рисунок 3.12 Допоміжна функція для завантаження зображень

Функція приймає на вхід датафрейм та набір даних. Ініціалізуємо список зображень, що буде заповнений побудованими фотомонтажами. Проходимося

циклом по будинках у датафреймі та беремо шлях до чотирьох зображень поточного будинку. Досягнуто першу поставлену вище мету – завантаження чотирьох зображень для кожного будинку. Усередині циклу виконаємо ініціалізацію списку, що буде містити чотири фотографії кожного запису (рис. 3.13). Для кожного зображення з набору завантажимо, змінимо розмір та додамо кожну фотографію до списку.

```
70. # initialize our list of input images along with the output image
71. # after *combining* the four input images
72. inputImages = []
73. outputImage = np.zeros((64, 64, 3), dtype="uint8")
74.
75. # loop over the input house paths
76. for housePath in housePaths:
77.     # load the input image, resize it to be 32 32, and then
78.     # update the list of input images
79.     image = cv2.imread(housePath)
80.     image = cv2.resize(image, (32, 32))
81.     inputImages.append(image)
82.
83. # tile the four input images in the output image such the first
84. # image goes in the top-right corner, the second image in the
85. # top-left corner, the third image in the bottom-right corner,
86. # and the final image in the bottom-left corner
87. outputImage[0:32, 0:32] = inputImages[0]
88. outputImage[0:32, 32:64] = inputImages[1]
89. outputImage[32:64, 32:64] = inputImages[2]
90. outputImage[32:64, 0:32] = inputImages[3]
91.
92. # add the tiled image to our set of images the network will be
93. # trained on
94. images.append(outputImage)
95.
96. # return our set of images
97. return np.array(images)
```

Рисунок 3.13 Обробка зображень

З чотирьох зображень для кожного будинку створено монтаж, що має наступну структуру:

- Зображення ванної у верхньому лівому куті;
- Зображення спальні у верхньому правому куті;
- Фронтальний вигляд знизу праворуч;
- Кухня знизу ліворуч.

Подібно до того, як числові та категоріальні атрибути представляють будинок, ці чотири фотографії, об'єднані в одне зображення, представлятимуть візуальну естетику будинку.

3.4. Проектування мережі для обробки числових та категоріальних даних

Категоріальні та числові дані обробляються простим багат шаровим перцептроном (рис. 3.14), що має наступну структуру:

- Повнозв'язний вхідний шар (Dense), функція активації – ReLU;
- Повнозв'язний прихований шар, функція активації – ReLU.

```

1. # import the necessary packages
2. from tensorflow.keras.models import Sequential
3. from tensorflow.keras.layers import BatchNormalization
4. from tensorflow.keras.layers import Conv2D
5. from tensorflow.keras.layers import MaxPooling2D
6. from tensorflow.keras.layers import Activation
7. from tensorflow.keras.layers import Dropout
8. from tensorflow.keras.layers import Dense
9. from tensorflow.keras.layers import Flatten
10. from tensorflow.keras.layers import Input
11. from tensorflow.keras.models import Model
12.
13. def create_mlp(dim, regress=False):
14.     # define our MLP network
15.     model = Sequential()
16.     model.add(Dense(8, input_dim=dim, activation="relu"))
17.     model.add(Dense(4, activation="relu"))
18.
19.     # check to see if the regression node should be added
20.     if regress:
21.         model.add(Dense(1, activation="linear"))
22.
23.     # return our model
24.     return model

```

Рисунок 3.14 Ініціалізація багатошарового персептрона

3.5. Проектування згорткової мережі

Зображення будинків представляють невидимий ресурс для прогнозування цін на будинки. Визначимо верхню праву гілку нашої моделі, а саме згорткову нейронну мережу. Використаємо заздалегідь побудовану модель. Кожна попередньо побудована модель в пакеті Keras має функцію попередньої обробки зображень для подальшого використання в моделі. Застосуємо цю функцію до зображень і використаємо їх як вхідні дані для моделей. Використання Keras та попередньо навчених моделей для вилучення ознак вимагає видалення остаточного (класифікаційного) шару кожної моделі. Це досягнуто шляхом встановлення параметра `include_top` значення `false` для конструктора об'єкта моделі.

Keras містить декілька популярних архітектур, для нашої задачі було обрано дослідити ResNet50 та VGG16, оскільки вони використовують дані ImageNet для попередньої підготовки, і тому деякі домени перекриваються з нашими зображеннями житла. Крім того, вони мають досить компактну архітектуру і довели свою успішність у змаганнях з класифікації зображень.

У табл. 3.2 наведено порівняння кількості вилучених ознак кожної з вибраних моделей. ResNet50 повертає 2048 функцій, що значно менше, ніж модель VGG. Зі

збільшенням кількості ознак за певний момент точність моделей машинного навчання, як правило, зменшується. Однак враховуючи результати попередніх досліджень, вирішено обрати VGG16.

Таблиця 3.2

Кількість параметрів та вилучених ознак попередньо тренованими моделями

Модель	Кількість ознак	Кількість параметрів
VGG16	25 088	138 357 544
ResNet50	2048	25 636 712

На рис. 3.15 представлені дві архітектури: оригінальна архітектура мережі VGG16, яка виводить ймовірності для кожної з 1000 міток класу ImageNet та мережа з видаленим останнім FC шаром. Замість повернення ймовірності для кожної з 1000 міток класу, мережа повертає вектор витягнутих ознак.

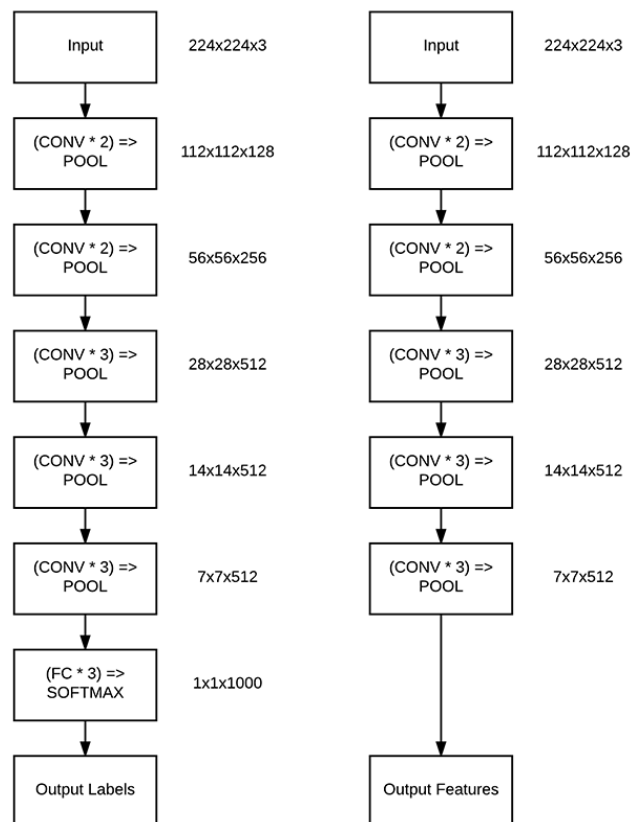


Рисунок 3.15 Оригінальна модель та модель для трансферного навчання

Ініціалізуємо мережу VGG16 (рис. 3.16) та будемо використовувати вихідні значення мережі безпосередньо, зберігаючи результати як вектори ознак.

```

1. # import the necessary packages
2. from sklearn.preprocessing import LabelEncoder
3. from tensorflow.keras.applications import VGG16
4. from tensorflow.keras.applications.vgg16 import preprocess_input
5. from tensorflow.keras.preprocessing.image import img_to_array
6. from tensorflow.keras.preprocessing.image import load_img
7. from pyimagesearch import config
8. from imutils import paths
9. import numpy as np
10. import pickle
11. import random
12. import os
13.
14. # load the VGG16 network and initialize the label encoder
15. print("[INFO] loading network...")
16. model = VGG16(weights="imagenet", include_top=False)
17. le = None

```

Рисунок 3.16 Ініціалізація згорткової нейронної мережі

3.6. Побудова фінальної моделі

Створимо фінальну модель, здатну обробляти як кілька входів, так і змішані дані, а саме об'єднаємо гілки та проведемо навчання та тестування створеної мережі. Імпортуємо усі необхідні пакети (рис. 3.17), а саме:

- **datasets**: три функції для завантаження та обробки даних CSV та завантаження і попередньої обробки фотографій будинку з набору даних будинків;
- **models**: Гілки MLP та CNN, що представляють собою змішані дані з кількома входами;
- **train_test_split**: функція scikit-learn для поділу даних для навчання та тестування мережі;
- **concatenate**: функція Keras, що дозволяє приймати кілька входів;
- **argparse**: обробляє аргументи командного рядка.

```

1. # import the necessary packages
2. from pyimagesearch import datasets
3. from pyimagesearch import models
4. from sklearn.model_selection import train_test_split
5. from tensorflow.keras.layers import Dense
6. from tensorflow.keras.models import Model
7. from tensorflow.keras.optimizers import Adam
8. from tensorflow.keras.layers import concatenate
9. import numpy as np
10. import argparse
11. import locale
12. import os
13.
14. # construct the argument parser and parse the arguments
15. ap = argparse.ArgumentParser()
16. ap.add_argument("-d", "--dataset", type=str, required=True,
17.                 help="path to input dataset of house images")
18. args = vars(ap.parse_args())

```

Рисунок 3.17 Імпорт пакетів для побудови фінальної моделі

Після цього виконаємо завантаження числових/категоріальних, а також даних зображень та масштабовано їх до діапазону $[0, 1]$ (рис. 3.18).

```
20. # construct the path to the input .txt file that contains information
21. # on each house in the dataset and then load the dataset
22. print("[INFO] loading house attributes...")
23. inputPath = os.path.sep.join([args["dataset"], "HousesInfo.txt"])
24. df = datasets.load_house_attributes(inputPath)
25.
26. # load the house images and then scale the pixel intensities to the
27. # range [0, 1]
28. print("[INFO] loading house images...")
29. images = datasets.load_house_images(df, args["dataset"])
30. images = images / 255.0
```

Рисунок 3.18 Завантаження даних

Датасет поділено на тренувальний та тестувальний у відношенні 75% та 25% відповідно (рис. 3.19). Знаходимо максимальну ціну у навчальному наборі і відповідно масштабуємо дані навчання та тестування. Розміщення даних про ціни в межах $[0, 1]$ призводить до кращого навчання та конвергенції.

```
32. # partition the data into training and testing splits using 75% of
33. # the data for training and the remaining 25% for testing
34. print("[INFO] processing data...")
35. split = train_test_split(df, images, test_size=0.25, random_state=42)
36. (trainAttrX, testAttrX, trainImagesX, testImagesX) = split
37.
38. # find the largest house price in the training set and use it to
39. # scale our house prices to the range [0, 1] (will lead to better
40. # training and convergence)
41. maxPrice = trainAttrX["price"].max()
42. trainY = trainAttrX["price"] / maxPrice
43. testY = testAttrX["price"] / maxPrice
44.
45. # process the house attributes data by performing min-max scaling
46. # on continuous features, one-hot encoding on categorical features,
47. # and then finally concatenating them together
48. (trainAttrX, testAttrX) = datasets.process_house_attributes(df,
49. trainAttrX, testAttrX)
```

Рисунок 3.19 Поділ датасету та масштабування цін

Об'єднаємо дві гілки мережі та сформуємо остаточну модель з кількома входами та єдиним виходом (рис. 3.20). Створюємо моделі MLP та CNN, після об'єднання їх виходів отримуємо комбінований вхід для решти мережі. Приєднуємо повнозв'язний шар до об'єданого входу. Додаємо шар з лінійною функцією активації, вихід якого – прогнозована ціна. Отримана модель отримує на вхід дві гілки, як багатовхідний сигнал та повертає одне значення – прогнозовану ціну.

Скомпілюємо та проведемо навчання моделі (рис. 3.21). Функція втрат – середнє абсолютне відхилення у відсотках, оптимізатор – “Adam”, кількість епох – 200, розмір батча – 8.

```

51. # create the MLP and CNN models
52. mlp = models.create_mlp(trainAttrX.shape[1], regress=False)
53. cnn = models.create_cnn(64, 64, 3, regress=False)
54.
55. # create the input to our final set of layers as the *output* of both
56. # the MLP and CNN
57. combinedInput = concatenate([mlp.output, cnn.output])
58.
59. # our final FC layer head will have two dense layers, the final one
60. # being our regression head
61. x = Dense(4, activation="relu")(combinedInput)
62. x = Dense(1, activation="linear")(x)
63.
64. # our final model will accept categorical/numerical data on the MLP
65. # input and images on the CNN input, outputting a single value (the
66. # predicted price of the house)
67. model = Model(inputs=[mlp.input, cnn.input], outputs=x)

```

Рисунок 3.20 Формування кінцевої моделі

```

69. # compile the model using mean absolute percentage error as our loss,
70. # implying that we seek to minimize the absolute percentage difference
71. # between our price *predictions* and the *actual prices*
72. opt = Adam(lr=1e-3, decay=1e-3 / 200)
73. model.compile(loss="mean_absolute_percentage_error", optimizer=opt)
74.
75. # train the model
76. print("[INFO] training model...")
77. model.fit(
78.     x=[trainAttrX, trainImagesX], y=trainY,
79.     validation_data=([testAttrX, testImagesX], testY),
80.     epochs=200, batch_size=8)
81.
82. # make predictions on the testing data
83. print("[INFO] predicting house prices...")
84. preds = model.predict([testAttrX, testImagesX])

```

Рисунок 3.21 Компілювання та тренування моделі

Для оцінки моделі обчислимо абсолютну різницю у відсотках і використаємо її для виведення кінцевих показників (рис. 3.22). Ці показники (середнє значення ціни, стандартне відхилення ціни та середнє значення + стандартне відхилення абсолютної різниці у відсотках) виведемо у терміналі.

```

86. # compute the difference between the *predicted* house prices and the
87. # *actual* house prices, then compute the percentage difference and
88. # the absolute percentage difference
89. diff = preds.flatten() - testY
90. percentDiff = (diff / testY) * 100
91. absPercentDiff = np.abs(percentDiff)
92.
93. # compute the mean and standard deviation of the absolute percentage
94. # difference
95. mean = np.mean(absPercentDiff)
96. std = np.std(absPercentDiff)
97.
98. # finally, show some statistics on our model
99. locale.setlocale(locale.LC_ALL, "en_US.UTF-8")
100. print("[INFO] avg. house price: {}, std house price: {}".format(
101.     locale.currency(df["price"].mean(), grouping=True),
102.     locale.currency(df["price"].std(), grouping=True)))
103. print("[INFO] mean: {:.2f}%, std: {:.2f}%".format(mean, std))

```

Рисунок 3.22 Оцінювання моделі

Середня абсолютна похибка у відсотках на початку тренування є дуже високо, але поступово знижується протягом навчального процесу (рис. 3.23). До кінця навчання отримано середню абсолютну відсоткову похибку на тестовому наборі 17,52%.

26.	...
27.	Epoch 191/200
28.	34/34 [=====] - 0s 4ms/step - loss: 23.4761 - val_loss: 23.4792
29.	Epoch 192/200
30.	34/34 [=====] - 0s 5ms/step - loss: 21.5748 - val_loss: 22.8284
31.	Epoch 193/200
32.	34/34 [=====] - 0s 3ms/step - loss: 21.7873 - val_loss: 23.2362
33.	Epoch 194/200
34.	34/34 [=====] - 0s 6ms/step - loss: 22.2006 - val_loss: 21.4601
35.	Epoch 195/200
36.	34/34 [=====] - 0s 3ms/step - loss: 22.1863 - val_loss: 23.8873
37.	Epoch 196/200
38.	34/34 [=====] - 0s 4ms/step - loss: 23.6857 - val_loss: 29.7415
39.	Epoch 197/200
40.	34/34 [=====] - 0s 4ms/step - loss: 19.0267 - val_loss: 26.4044
41.	Epoch 198/200
42.	34/34 [=====] - 0s 4ms/step - loss: 17.7724 - val_loss: 23.4979
43.	Epoch 199/200
44.	34/34 [=====] - 0s 3ms/step - loss: 17.1597 - val_loss: 19.2382
45.	Epoch 200/200
46.	34/34 [=====] - 0s 3ms/step - loss: 16.9746 - val_loss: 17.5241
47.	[INFO] predicting house prices...
48.	[INFO] avg. house price: \$230,388.27, std house price: \$210,712.08
49.	[INFO] mean: 17.52%, std: 22.19%

Рисунок 3.23 Результати тестування мережі

3.7. Опис функціональності веб-додатку

Для взаємодії з кінцевим користувачем створено веб-додаток “Rieltron”. На рис. 3.24 зображений екран входу.

Рисунок 3.24 Екран входу

Для того, щоб увійти в систему користувачу необхідно ввести пошту та пароль. Для нових користувачів на екрані є кнопка “Зареєструватися”. Після того як користувач увійшов до системи він потрапляє на сторінку вводу даних (3.25).

The screenshot shows the REALTRON web application interface. At the top, there is a yellow header with the REALTRON logo and the tagline "Інтелектуальна система оцінювання вартості житла". A "Вийти" (Logout) button is in the top right corner. Below the header, there are three tabs: "Характеристики" (Characteristics), "Зображення" (Images), and "Оцінка вартості" (Value Assessment). The "Характеристики" tab is active. It contains four input fields: "Кількість спалень *" (Number of bedrooms) with the value 3, "Кількість ванних кімнат" (Number of bathrooms) with the value 1, "Загальна площа" (Total area) with the value 82, and "Поштовий індекс" (Postal index) with the value 40021 and a clear button (X).

Рисунок 3.25 Екран вводу характеристик будинку

Після введення кількості спалень, кількості ванних кімнат, загальної площі будинку та поштового індексу, користувачу необхідно завантажити зображення будинку. Для цього потрібно перейти на вкладку “Зображення” (рис. 3.26).

The screenshot shows the REALTRON web application interface with the "Зображення" (Images) tab active. It displays four photo upload slots arranged in a 2x2 grid. The top-left slot is labeled "Фото спальні" (Bedroom photo) and shows a bedroom interior. The top-right slot is labeled "Фото ванної кімнати" (Bathroom photo) and shows a bathroom interior. The bottom-left slot is labeled "Фото кухні" (Kitchen photo) and shows a kitchen interior. The bottom-right slot is labeled "Фронтальний вид на будинок" (Front view of the house) and shows the exterior of a house. Each slot has a small thumbnail image and a larger empty area for the main photo.

Рисунок 3.26 Екран завантаження зображень

На вкладці “Зображення” користувачу потрібно завантажити фото спальні, фото кухні, фото ванної кімнати та фронтальний вид на будинок. Після заповнення усіх полів на вкладках “Характеристики” та “Зображення” та перейшовши на вкладку “Оцінка вартості”, користувач отримує прогнозовану ціну будинку (рис. 3.27).

Рисунок 3.27 Екран отримання оцінки вартості житла

Висновки до розділу

У даному розділі спроектовано нейронну мережу, здатну приймати на вхід кілька типів даних одночасно, а саме: числові, категорійні та дані зображення. Числові та категоріальні дані об'єднано в єдиний вектор ознак, щоб сформувати перший вхід до мережі, дані зображень є другим входом до мережі. Модель складається з двох гілок: перша гілка моделі є багатошаровим персептроном і містить строго пов'язані шари (для об'єднаних числових та категоріальних даних), тоді як друга гілка багатовхідної моделі є згортковою нейронною мережею VGG16. Виходи обох гілок об'єднуються та визначають єдиний результат – передбачення регресії.

Сформовано датасет, що складається з 2140 зразків зображень будинків, що були виставлені на продаж на території України під час формування датасету та текстового файлу з текстовими метаданими набору даних. Датасет поділено на

тренувальний та тестувальний у відношенні 75% та 25% відповідно. Проведено навчання єдиної мережі на змішаних даних. Проаналізовано та оцінено точність прогнозування моделі з кількома входами та різними типами вхідних даних. Отримана модель на тестових даних демонструє точність на рівні 82,48%.

Для взаємодії з кінцевим користувачем створено веб-додаток та детально описано його функціональність.

РОЗДІЛ 4. МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП-ПРОЕКТУ

За даними CB Insights, ринкової та бізнес-аналітичної платформи, однією з основних причин, через які стартап-проекти провалюються є те, що ринок не потребує даного продукту. Інші важливі проблеми – брак капіталу, відсутність потрібної команди, проблеми з встановленням цін, поганий маркетинг тощо. Численні засновники стартапів стверджують, що основною причиною невдачі все ще є недооцінка вартості дослідження ринку з самого початку. Найкраще починати з маркетингового аналізу стартапа перед написанням бізнес-плану. Дослідження дозволяє проаналізувати, які продукти є прибутковими, а які не відповідають вимогам ринку. Необхідно ознайомитись із основною аудиторією, конкурентами та галуззю в цілому, щоб визначити справжню цінність товару.

Дослідження ринку створює міцну основу для розвитку бізнесу, готуючи компанію до будь-яких загроз або слабких місць, які можуть виникнути в міру зростання бізнесу. Від знайомства з цільовою аудиторією до вивчення потенційних конкурентів, дослідження ринку надає бізнесу конкурентну перевагу, дозволяючи їм процвітати в нових середовищах. Загалом виділяють чотири головні етапи аналізу стартапу:

1. Маркетинговий аналіз стартапу

На цьому етапі необхідно описати ідею стартапу, його мету та те, як його можна використовувати. Також необхідно здійснити огляд вже існуючих рішень і визначити їх переваги та недоліки. Відкриваючи новий бізнес, обов'язково потрібно знати про будь-які компанії, які можуть конкурувати. Незалежно від того, чи це прямий чи непрямий конкурент, знання того, як їхній бізнес може загрожувати вашому, допоможе краще просувати власний продукт і залишатися конкурентоспроможним серед спільної аудиторії. Під час маркетингового аналізу значну увагу приділяють аналізу ринку та методам виведення товару на ринок.

2. Організація стартапу

Цей етап передбачає проведення аналізу щодо необхідних ресурсів як матеріальних, так і нематеріальних. Проводяться розрахунки необхідних витрат, що будуть на початковому етапі. Перш ніж почнеться розробка стартапу, потрібно

буде оплатити рахунки. Розуміння витрат допомагає успішно запустити розробку. Розрахунок стартових витрат допомагає оцінити прибуток, зробити аналіз беззбитковості, брати безпечні позики, залучити інвесторів. Інвестори та позикодавці порівнюють очікувані витрати з прогнозованим доходом та визначають потенціал бізнесу для отримання прибутку. Також складають приблизний календарний план робіт.

3. Фінансово-економічний аналіз проекту та оцінка ризиків

Під час фінансово-економічного аналізу проекту проводять розрахунки основних економічних показників, визначають інвестиційну привабливість проекту. Під час цього етапу також визначаються потенційні ризики та шляхи їх попередження. Це може включати такі речі, як нові конкуренти, зміни в законодавстві, фінансові ризики та практично все інше, що потенційно може поставити під загрозу майбутнє проекту.

4. Комерціалізація проекту

Даний етап передбачає побудову стратегії щодо пошуку інвесторів для стартапу та визначає цільову групу інвесторів. Швидкий пошук інвесторів не є простим завданням. Більшість венчурних капіталістів отримують десятки запитів щодня і просто не встигають зустрітися з усіма. Тому цей процес є дуже важливим та потребує попередньої підготовки.

4.1. Опис ідеї проекту

Мета роботи полягає у тому, щоб розробити систему, що базується на нейронній мережі і вирішує задачу оцінювання ринкової вартості житла. На основі даного рішення створити веб-додаток для взаємодії з кінцевим користувачем. Сформуємо зміст ідеї та розглянемо детально основні напрямки застосування та вигоду для користувачів (табл. 4.1).

Розглянемо застосунки, що вже створені для оцінювання ринкової вартості житла. Експерти стверджують, що аналіз ринку життєво важливий на самому ранньому етапі створення стартапу. Вузьке розуміння ділового середовища або відсутність даних про конкурентні переваги є одними з найпоширеніших помилок у презентаціях для інвесторів. Отже, якщо для того, щоб залучити інвестора,

найбільше уваги слід приділити вивченню ринку під час створення бізнес-плану. Отримані дані повинні бути нещодавніми, інформативними та точними. Проведемо ретельний аналіз продуктів конкурентів та визначимо сильні і слабкі сторони (табл. 4.2).

Таблиця 4.1

Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигода для користувача
Розробка системи для оцінювання ринкової вартості житла	<ul style="list-style-type: none"> – Визначення ринкової вартості житла для купівлі/продажу без залучення агентів з нерухомості; – Визначення оціночної вартості житла при наданні позик під заставу нерухомості; – Визначення ринкової вартості житла страховими компаніями. 	<ul style="list-style-type: none"> – Зменшення витрат при операціях купівлі та продажу житла; – Пришвидшення процесу виставлення житла на продаж; – Оцінювання справедливості виставленої ціни; – Пришвидшення процесу надання;

Таблиця 4.2

Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№	Техніко-економічні характеристики ідеї	Продукція конкурентів					W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Моя система	DOM.RIA	Zestimate	RE/MAX	Redfin			
1	Оцінювання ринкової вартості житла в Україні	+	+	-	-	-			+
2	Точність результатів	+	-	+	+	+		+	
3	Швидкодія системи	+	+	+	-	+		+	

4	Веб-додаток	+	+	+	+	+		+	
5	Зручність використання	+	-	-	-	+			+
6	Використання нейромережі	+	-	+	-	-			+
7	Підтримка української мови	+	+	-	-	-			+
8	Використання даних зображень	+	-	+	-	-			+

Проведено аналіз вже існуючих систем та визначено слабкі та сильні сторони стартапу. Головними недоліками існуючих рішень є те, що більшість додатків не можуть бути використані для оцінювання нерухомості в Україні і те, що приблизна оцінка розраховується, базуючись лише на цінах будинків, що були продані у тому ж районі, тобто жодні інші характеристики не враховуються. Хоча ці оцінки часто близькі до фактичних цін продажу, в деяких випадках вони є дуже неточними.

4.2. Технологічний аудит ідеї проекту

Розглянемо та проаналізуємо технології, що необхідні для реалізації ідеї стартапу. Зокрема визначимо наявність та доступність технологій (табл. 4.3).

Таблиця 4.3

Технологічна здійсненність проекту

№	Ідея проекту	Технології реалізації	Наявність технологій	Доступність технологій
1	Нейронна мережа	Tensorflow, Keras, Python	+	+
2	Web-додаток	JavaScript, NodeJS, Express, NestJS, Angular	+	+

4.3. Аналіз ринкових можливостей запуску стартап-проекту

При першому дослідженні ринку важливо з'ясувати, які компанії працюють у вашій ніші. І не менш важливо зрозуміти, як ці компанії потрапили до списку

найкращих гравців галузі та яких маркетингових стратегій вони дотримуються. Також потрібно дослідити характеристики та уподобання цільової аудиторії, регіональні можливості, пропозиції конкурентів та загальний динамічний розвиток ринку. Більшість власників стартапів воліють перевіряти інші малі підприємства. Здається очевидним досліджувати компанії, які тільки починають, що працюють з подібним бюджетом, і не можуть похвалитися надзвичайною впізнаваністю бренду. Засновники сподіваються, що це допоможе їм перевершити найближчих суперників у тактиці і, в якийсь момент, це справді так.

Однак така тактика не сприяє довгостроковій стратегії. Для стійкого зростання необхідний інший підхід. Стартапам може бути вигідніше досліджувати нові горизонти, аналізуючи найкращі практики успішного бізнесу. Вивчення їх досвіду може заощадити час та ресурси та з меншими зусиллями привести до поставленої мети. Проведемо аналіз впровадження та виведення даного стартап-проекту на ринок (табл. 4.4).

Таблиця 4.4

Попередня характеристика потенційного ринку

№ п/п	Показники стану ринку	Характеристика
1	Кількість головних гравців, од	4
2	Загальний обсяг продаж, грн/ум.од	100000
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Відсутні
5	Специфічні вимоги до стандартизації та сертифікації	Відсутні
6	Середня норма рентабельності в галузі (або по ринку), %	50

За допомогою аналізу потенційного ринку встановлено, що ринок є привабливим. Це зумовлено невеликою кількістю конкурентів, динамікою ринку, що зростає та високою рентабельністю.

Перед початком запуску стартапу необхідно встановити, хто є потенційними клієнти та як їх залучити. Цільовий ринок – це не всі, хто знаходиться у районі вашого бізнесу. Він складається лише з людей, які добре підходять для запропонованого продукту. Потрібно точно визначити, хто ваші клієнти і де їх

знайти. Ця розбивка називається сегментацією цільового ринку. визначити цільову аудиторію стартап-проекту (табл. 4.5).

Таблиця 4.5

Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Можливість оцінити житло для продажу/купівлі	Звичайні користувачі	Регіональне розташування, клас нерухомості, що оцінюється	Висока точність, інтуїтивна зрозумілість, зручність користування, швидкість
2	Можливість оцінити вартість житла для надання кредитів під заставу	Банківські установи	Розмір установи	Зацікавлені у дуже високій точності
3	Можливість оцінити вартість житла для страхування	Страхові компанії	Розмір установи	Зацікавлені у дуже високій точності

Встановлено, що потенційними клієнтами стартапу є звичайні користувачі, банківські установи та страхові компанії.

Можливості – це, як правило, зовнішні фактори організації, які мають велику ймовірність досягнення успіху. Можливості завжди супроводжуються загрозами. Це перешкоджаючі фактори, що неможливо контролювати. Єдиний спосіб протистояти цьому – розробити надійний план дій на випадок непередбачених ситуацій, щоб ефективно боротися з ними, коли вони трапляються. Можна скористатися можливостями та захиститися від загроз, але не можливо їх змінити.

Зробимо аналіз ринкового середовища, а саме визначимо фактори основних загроз, що можуть вплинути на продаж розробленої системи (табл. 4.6). Також визначимо фактори можливостей (табл. 4.7), які можуть вплинути на розроблений продукт під час виходу його на ринок.

Таблиця 4.6

Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Кібератака	Перевантаження серверів веб-додатку, блокування системи	Впровадити потужну навчальну програму з кібербезпеки для працівників, впровадити технологію брандмауера наступного покоління, впровадити засоби виявлення та захисту від ботів
2	Вихід конкурентів на ринок	Після виходу продукту на ринок можуть з'явитися додатки, що вирішують таку ж проблему	Розробка нових функцій додатку
3	Недостатнє фінансування	Відсутність коштів	Пошук нових інвесторів, розширення клієнтської бази

Таблиця 4.7

Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Відсутність аналогів	В Україні немає аналогів такого програмного продукту	Можливість стати монополістом в даній галузі в Україні
2	Монетизація додатку	Впровадження контекстної та таргетованої реклами у додаток	Отримання додаткового прибутку
3	Зацікавлення представників з інших сфер діяльності	Адаптування веб-додатку під інші сфери діяльності (Оцінка потенційної вартості ремонту у житловому приміщенні)	Розширення цільової аудиторії

Проведемо ступеневий аналіз конкуренції на ринку (табл. 4.8). Даний аналіз допомагає проаналізувати особливості конкурентного середовища. На базі цього можна побудувати основну тактику залучення користувачів.

Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Тип конкуренції: Олігополія	Існують кілька подібних програмних засобів, але їх небагато. В Україні конкуренти відсутні.	Взяти усе найкраще у конкурентів, впровадити свої покращення.
2. За рівнем конкурентної боротьби: Міжнаціональний	Є декілька конкурентів в інших країнах.	Ці програми не мають поширення в Україні та не можуть бути застосовані для оцінювання ринкової вартості саме українського житла.
3. За галузевою ознакою: Внутрішньогалузева	Боротьба серед аналогів за клієнта	Необхідно створити кращий товар ніж у конкурентів
4. Конкуренція за видами товарів: Товарно-видова	Товари одного виду	Розробити якісне ПЗ та створити активну рекламну кампанію
5. За характером конкурентних переваг: Нецінова	Програмне забезпечення буде більш зручним, а також буде характеризуватися високою точністю результатів	Необхідно досягти високої точності роботи нейронної мережі, а також слідувати UX гайдам під час розробки додатку
6. За інтенсивністю: Не марочна	Фірма ще не набула достатньої популярності	Створення дійсно якісного продукту для завоювання ринку

Аналіз конкурентів за Портером – найкорисніший інструмент для власників та менеджерів, щоб бути на крок попереду конкурентів на складних ринках (табл. 4.9). Це аналітична модель, яка допомагає маркетологам та керівникам підприємств подивитися на “баланс сил” на ринку між різними організаціями на глобальному рівні та проаналізувати привабливість та потенційну прибутковість галузевого сектору.

Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	DOM.RIA, Zestimate, RE/Max, Redfin	Юридична частина, розробка	-	Зручний інтерфейс	Часткова заміна
Висновки	Дуже інтенсивна	Є потенційні конкуренти на території інших держав	-	Умови до ПЗ	Товари замінники частково присутні, планується перекрити їх якістю та рекламною кампанією.

Проаналізуємо та розглянемо фактори конкурентоспроможності (табл. 4.10). Наведемо чинники, що роблять даний фактор значущим для порівняння конкурентів та порівняємо з конкурентами (табл. 4.11).

Таблиця 4.10

Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Ціна	Додатки конкурентів є безкоштовними для звичайних користувачів.
2	Легкість у користуванні	Інтерфейс повинен бути інтуїтивно зрозумілим для того, щоб закріпитися на ринку.
3	Задоволення потреб клієнтів	Клієнти не мають широкого вибору додатків для автоматичного оцінювання вартості житла.
4	Висока точність	Додаток повинен повертати результат з високою точністю.
5	Висока якість продукту	Якісний продукт може швидше зайти на ринок та закріпитися на ньому.
6	Врахування поточного стану та ремонту житла	Даний фактор значно підвищує точність. Не всі конкуренти мають підтримку оцінювання житла за зображенням.

Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з запропонованим товаром						
			-3	-2	-1	0	+1	+2	+3
1	Ціна	15				+			
2	Легкість у користуванні	18			+				
3	Задоволення потреб клієнтів	18		+					
4	Висока точність	20			+				
5	Висока якість продукту	18				+			
6	Врахування поточного стану та ремонт житла	18		+					

Аналіз SWOT (сильних, слабких сторін, можливостей та загроз) – це основа, яка використовується для оцінки конкурентних позицій компанії та розробки стратегічного планування. SWOT-аналіз оцінює внутрішні та зовнішні фактори, а також поточний та майбутній потенціал. SWOT-аналіз допомагає максимально використати сильні сторони, з найкращою перевагою для організації (табл. 4.12). Він також може зменшити шанси на невдачу, так як виявляє те, чого не вистачає стартапу, і передбачає небезпеки, які в іншому випадку застали б зненацька.

Таблиця 4.12

SWOT-аналіз стартап-проекту

Сильні сторони: <ul style="list-style-type: none"> – Собівартість; – Зручність і простота користування; – Використання нейронної мережі; – Висока точність результатів; – Врахування поточного стану та ремонту житла; – Висока якість продукту. 	Слабкі сторони: <ul style="list-style-type: none"> – Відсутність чіткого вектору інтеграції з банками та страховими компаніями.
Можливості: <ul style="list-style-type: none"> – Відсутність аналогів; – Монетизація додатку; – Зацікавлення представників з інших сфер діяльності. 	Загрози: <ul style="list-style-type: none"> – Кібератака; – Недостатнє фінансування; – Поява конкурентів на українському ринку.

За допомогою отриманого SWOT-аналізу необхідно розглянути альтернативи ринкової поведінки (табл. 4.13).

Таблиця 4.13

Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Ефективна реклама, що зацікавить клієнтів	30%	3 місяці
2	Надання тимчасового доступу до додатку банківським установам	40%	6 місяців
3	Надання тимчасового доступу до додатку страховим компаніям	40%	6 місяців
4	Презентування товару на виставках, конференціях	50%	3 місяці

Обрано альтернативу “Презентування товару на виставках, конференціях”, оскільки за строками реалізації та ймовірністю отримання ресурсів вона є найбільш оптимізованою.

4.4. Розроблення ринкової стратегії проекту

Наявність ринкової стратегії є ключем до успіху будь-якого бізнесу. Маркетинг не є самотійною, одноразовою діяльністю. Він складається з кількох різних компонентів, необхідних на кожному етапі діяльності компанії – від задовго до того, як продаж навіть почнеться, до довгого після. Хоча ринкова стратегія може вимагати регулярних коригувань або доопрацювань, вона надає шаблон, з чого почати, і полегшує перегляд подібних або покращених результатів кожної кампанії без необхідності повністю починати з початку.

Визначення стратегії охоплення ринку передусім саме розробленню ринкової стратегії стартап-проекту. Оберемо цільові групи потенційних споживачів та проведемо їх аналіз (табл. 4.14).

Таблиця 4.14

Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Люди, що хочуть купити/продати житло	Готові	Дуже високий	Є декілька конкурентів	Просто
2	Банківські установи	Готові	Високий	Немає	Середня
3	Страхові компанії	Готові	Високий	Немає	Середня

На початковому етапі вирішено обрати як цільову групу – люди, що хочуть купити/продати житло. У подальшому можливе розширення цільової аудиторії на банківські установи та страхові компанії. Визначимо базову стратегію розвитку (табл. 4.15).

Таблиця 4.15

Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	Зосередження на одному сегменті	Стратегія концентровано го маркетингу	Відсутність великих ресурсів	Стратегія спеціалізації

Після цього необхідно визначити стратегію конкурентної поведінки (табл. 4.16), а також стратегію позиціонування (табл. 4.17). Стратегія конкурентної поведінки визначає активність та напрям дій по відношенню до лідера відповідного ринку. Урахування в явній формі позицій і дій конкурентів представляє важливий компонент стратегії розвитку. Основна мета стратегії позиціонування полягає в

тому, щоб виробити прихильність споживача до товару фірми на основі позитивних відмінностей цього товару від товарів конкурентів.

Таблиця 4.16

Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект першопрохідцем на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
1	Ні	Компанія буде шукати нових споживачів	Так , функціональні можливості	Стратегія виклику лідера

Таблиця 4.17

Визначення базової стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1	Зручність у використанні, висока точність результатів	Позиціонування за співвідношенням "ціна - якість"	Використання нейронної мережі, використання даних зображення, що значно підвищують точність результатів, зручний інтерфейс веб-додатку	Якість, простота використання, вітчизняний продукт

4.5. Розроблення маркетингової програми стартап-проекту

Маркетингова стратегія допомагає створювати товари та послуги з найбільшими шансами на отримання прибутку. Це тому, що маркетингова стратегія починається з дослідження ринку, беручи до уваги оптимального цільового клієнта, те, що роблять конкуренти та які тенденції можуть бути на горизонті. Використовуючи цю інформацію, необхідно визначити як можна

відрізнити свій товар чи послугу від конкурентів. Визначимо ключові переваги товару (табл. 4.18) та опишемо три рівня моделі товару (табл. 4.19).

Таблиця 4.18

Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Оцінювання вартості житла	Точність результатів	Висока точність оцінки ринкової вартості житла шляхом використання нейронних мереж та обробки даних різних типів (числових, категоріальних та даних зображення)

Таблиця 4.19

Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Інтелектуальна система оцінювання ринкової вартості житла на базі нейронної мережі, що здатна приймати на вхід дані різних типів, а саме: числові, категоріальні та дані зображення.		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. Точність оцінювання ринкової вартості житла		
	2. Швидкість роботи		
	3. Зрозумілість та простота використання		
	4. Підтримка української мови		
	Якість: вимоги до створення веб-додатку		
	Марка: Ріелтрон		
III. Товар із підкріпленням	Інтеграція з агенствами нерухомості, маркетплейсами для купівлі чи продажу житла. Інтеграція з банківськими установами та страховими компаніями. Можливість оцінювання вартості житла в цілях оренди.		
За рахунок чого потенційний товар буде захищено від копіювання: захист інтелектуальної власності			

Частиною маркетингової стратегії є встановлення правильної ціни на товар чи послугу, виходячи з даних дослідження ринку. Якщо клієнти хочуть елітного товару у вашій категорії, цінова стратегія може вимагати, щоб ви продавали за цінами, які створюють високу цінність. Якщо цільовий клієнт усвідомлює торг і готовий прийняти менше особливостей в продукті в обмін на меншу плату, цінова стратегія вимагатиме, щоб продавати товар за ціною конкуренції або нижче. Щоб вирішити, яка повинна бути ціна за товар потрібно більше продумати, ніж просто підрахувати витрати та додати націнку. Для цього проведемо аналіз та визначимо межі встановлення ціни (табл. 4.20), а також сформуємо систему збуту (табл. 4.21).

Таблиця 4.20

Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	Розробка додатку: мінімум \$5500	-	Вільно середній	Інтегрування в банківські та страхові системи: 10000 грн/місяць; Реклама агенств нерухомості: 500 грн/місяць.

Таблиця 4.21

Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Клієнт хоче отримати найкращий товар за співвідношенням ціна-якість	Залучення нових клієнтів	Нульовий рівень	Вертикальна система збуту

Стратегія маркетингових комунікацій – це стратегія, що використовується компанією чи приватною особою для досягнення цільового ринку за допомогою різних видів комунікацій. Перш ніж створити стратегічний план комунікацій необхідно зрозуміти свою цільову аудиторію. Будь-який план маркетингових

комунікацій повинен бути сформульований для певної групи цільових споживачів. Так як це було виконано вище, проведемо аналіз та визначимо концепцію маркетингових комунікацій (табл. 4.22).

Таблиця 4.22

Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Орієнтовані на високу точність результатів оцінювання та якість додатку в цілому	Контент-маркетинг	Висока точність	Показати якість продукту	Показати якість продукту

Висновки до розділу

У даному розділі проведено маркетинговий аналіз стартап-проекту. Зокрема проведено SWOT-аналіз, як результат визначено сильні та слабкі сторони, можливості та загрози проекту. Проаналізовані ринкові можливості запуску стартапу, за допомогою аналізу потенційного ринку встановлено, що ринок є привабливим. Це зумовлено невеликою кількістю конкурентів, динамікою ринку, що зростає та високою рентабельністю.

Потенційними клієнтами стартапу є звичайні користувачі, банківські установи та страхові компанії. Спільною потребою усіх груп користувачів є висока точність оцінки вартості житла. Проведено ступеневий аналіз конкуренції на ринку, визначені особливості конкурентів. Також проведено аналіз конкуренції за Портером та розглянуто фактори конкурентоспроможності. На базі цього побудовано основну тактику залучення користувачів. На початковому етапі вирішено обрати як цільову групу людей, що хочуть купити/продати житло. У подальшому можливе розширення цільової аудиторії на банківські установи та страхові компанії.

Визначено базову стратегію розвитку, стратегію конкурентної поведінки та стратегію позиціонування. Стратегією охоплення ринку обрано стратегію концентрованого маркетингу, а альтернативою розвитку проекту – зосередження на одному сегменті, адже на початковому етапі відсутня велика кількість ресурсів. Також визначені можливі способи монетизації додатку та встановлено ціну на товар. Розроблено концепцію маркетингових комунікацій.

Маркетинговий аналіз стартапу встановив, що розробка та впровадження проекту є доцільною.

ВИСНОВКИ

У даній магістерській дисертації розглянуто проблему оцінювання ринкової вартості житла та досліджено застосування методів машинного навчання для прогнозування ринкової вартості житла, зокрема можливість використання візуальної інформації для покращення прогнозів.

Проведено аналіз вже існуючих систем та визначено їх переваги та недоліки. Головними недоліками існуючих рішень є те, що більшість додатків не можуть бути використані для оцінювання нерухомості в Україні і те, що приблизна оцінка розраховується, базуючись лише на цінах будинків, що були продані у тому ж районі, тобто жодні інші характеристики не враховуються. За результатами проведеного аналізу визначено задачі, що повинні бути вирішені для того, щоб створити інтелектуальну систему оцінювання ринкової вартості житла, а також вимоги, яким повинна відповідати створена система.

Розглянуто методи вирішення задачі оцінювання ринкової вартості житла. Визначено загальний підхід до побудови системи, зокрема використання нейронної мережі. Вирішено навчити модель аналізувати образ будинку та враховувати інформацію щодо естетичної якості будинку при прогнозуванні ціни. Також обрано технології для розробки веб-додатку та нейронної мережі. Для розробки та тренування нейронної мережі вирішено використовувати мову програмування Python, бібліотеки машинного навчання TensorFlow, Keras, Scikit Learn, Pandas. Для розробки веб-додатку вирішено використовувати мову програмування JavaScript, Node.js, фреймворк Angular та бібліотеку UI компонентів Angular Material.

Спроековано нейронну мережу, здатну приймати на вхід кілька типів даних одночасно, а саме: числові, категоріальні та дані зображення. Модель складається з двох гілок: перша гілка моделі є багатошаровим персептроном, тоді як друга гілка багатовхідної моделі є згортковою нейронною мережею VGG16. Виходи обох гілок об'єднуються та визначають єдиний результат – передбачення регресії. Проведено навчання єдиної мережі на змішаних даних. Проаналізовано та оцінено точність прогнозування моделі з кількома входами та різними типами вхідних даних. Отримана модель на тестових даних демонструє точність на рівні 82,48%. Для

взаємодії з кінцевим користувачем створено веб-додаток та детально описано його функціональність.

Проведено маркетинговий аналіз стартап-проекту. Зокрема проведено SWOT-аналіз, як результат визначено сильні та слабкі сторони, можливості та загрози проекту. Потенційними клієнтами стартапу є звичайні користувачі, банківські установи та страхові компанії. Спільною потребою усіх груп користувачів є висока точність оцінки вартості житла. Маркетинговий аналіз стартапу встановив, що розробка та впровадження проекту є доцільною.

ПЕРЕЛІК ПОСИЛАНЬ

1. Raschka S. Python Machine Learning / Sebastian Raschka., 2015. – (1st Edition).
2. Flanagan D. JavaScript: The Definitive Guide / David Flanagan., 2011. – (O'Reilly Media). – (6th Edition) – 115p.
3. Géron A. Hands-On Machine Learning with Scikit-Learn and TensorFlow / Aurélien Géron., 2017. – (O'Reilly Media). – (1st Edition) – 110 p.
4. Elliott E. JavaScript Applications / Eric Elliott., 2014. – (O'Reilly Media). – (1st Edition) – 621 p.
5. Типи інстансів Amazon EC2 [Електронний ресурс] – Режим доступу до ресурсу: <https://aws.amazon.com/ru/ec2/instance-types/>.
6. House Prices Prediction Using Deep Learning [Електронний ресурс] – Режим доступу до ресурсу: <https://towardsdatascience.com/house-prices-prediction-using-deep-learning-dea265cc3154>.
7. The Multi-Channel Neural Network Learning [Електронний ресурс] – Режим доступу до ресурсу: <https://towardsdatascience.com/the-multi-channel-neural-network-26551bdfab6c>.
8. Transfer Learning [Електронний ресурс] – Режим доступу до ресурсу: <https://ruder.io/transfer-learning/>.
9. TensorFlow API Documentation [Електронний ресурс] – Режим доступу до ресурсу: https://www.tensorflow.org/api_docs.
10. Мова розмітки HTML [Електронний ресурс] – Режим доступу до ресурсу: <https://devdocs.io/html/>.
11. Програмна платформа Node.JS [Електронний ресурс] – Режим доступу до ресурсу: <https://nodejs.org/uk/docs>.
12. SWOT Analysis for Your Small Business [Електронний ресурс] – Режим доступу до ресурсу: <https://www.wordstream.com/blog/ws/2017/12/20/swot-analysis>.
13. How to do market research in 4 steps [Електронний ресурс] – Режим доступу до ресурсу: <https://www.hotjar.com/blog/market-research/>.

ДОДАТКИ

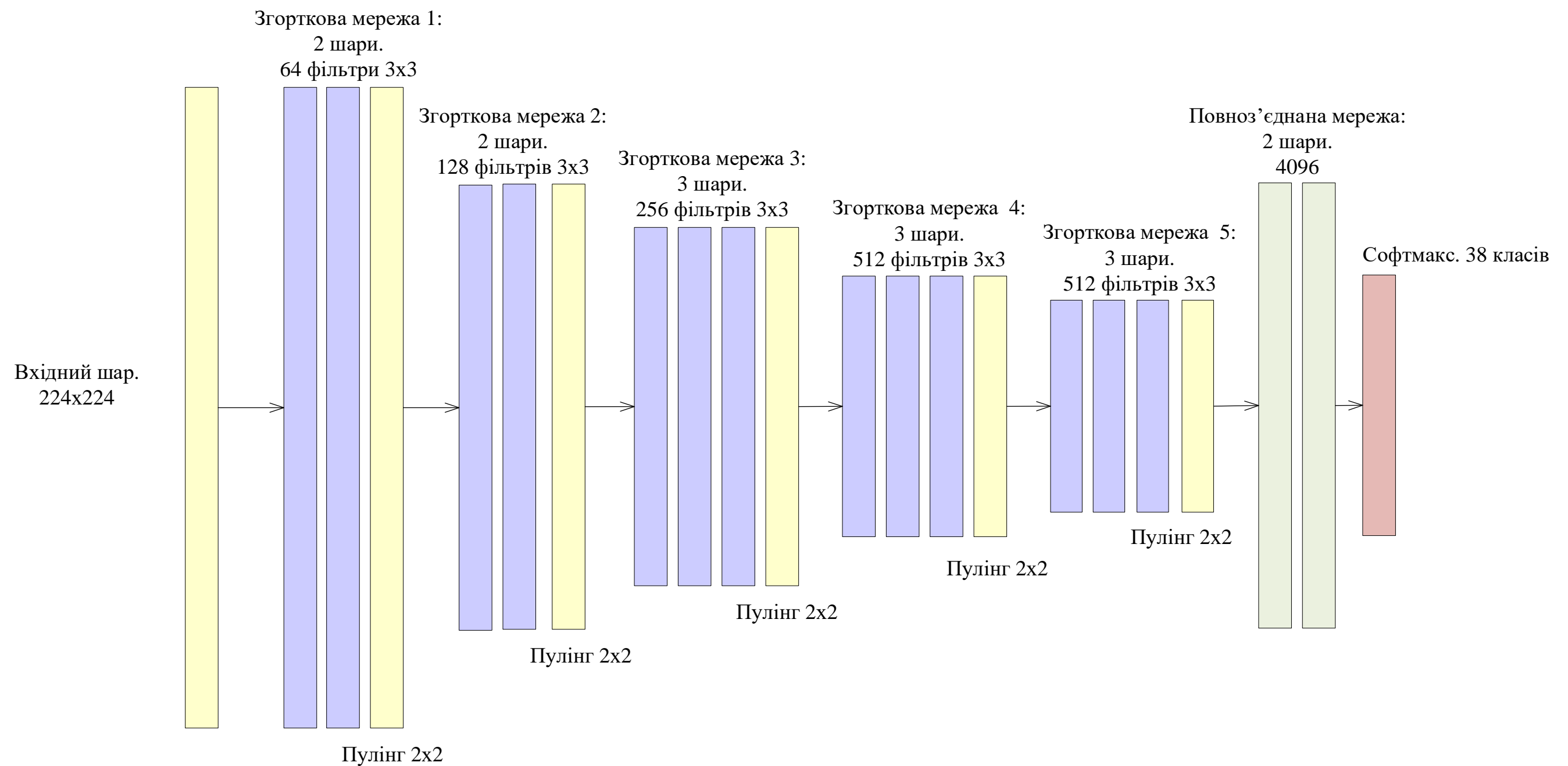
ДОДАТОК А

Графічні матеріали

ДОДАТОК Б

Результат перевірки на співпадіння

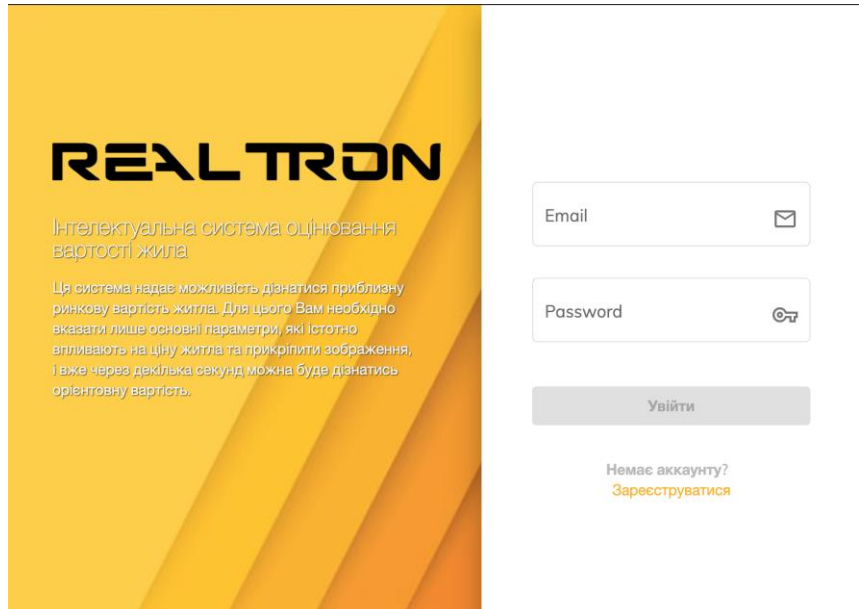
Архітектура нейронної мережі VGG16



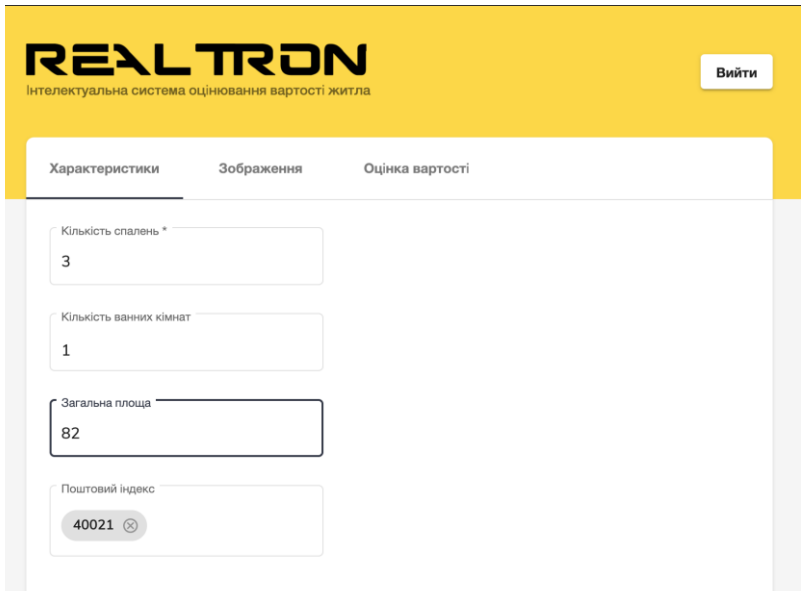
Демонстраційний плакат №1 «Архітектура нейронної мережі VGG16»
до дипломної роботи на тему
«Інтелектуальна система оцінювання ринкової вартості житла»

Виконала: студентка гр. ІК-91мп Велітченко Ю.В.
Керівник: д. т. н., доцент Корнага Я.І.

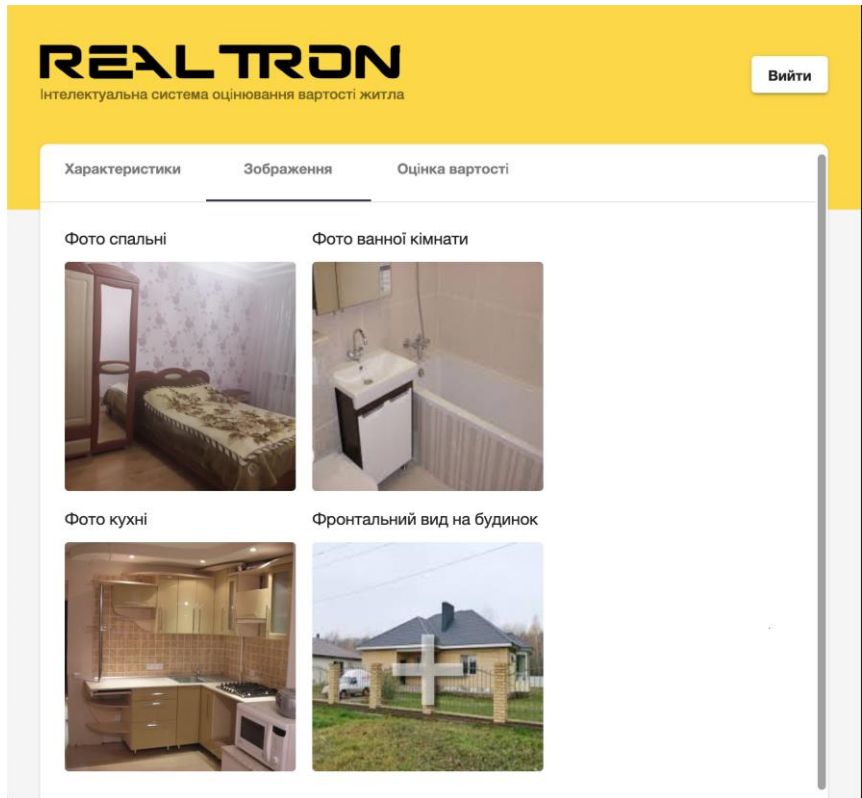
Інтерфейс користувача



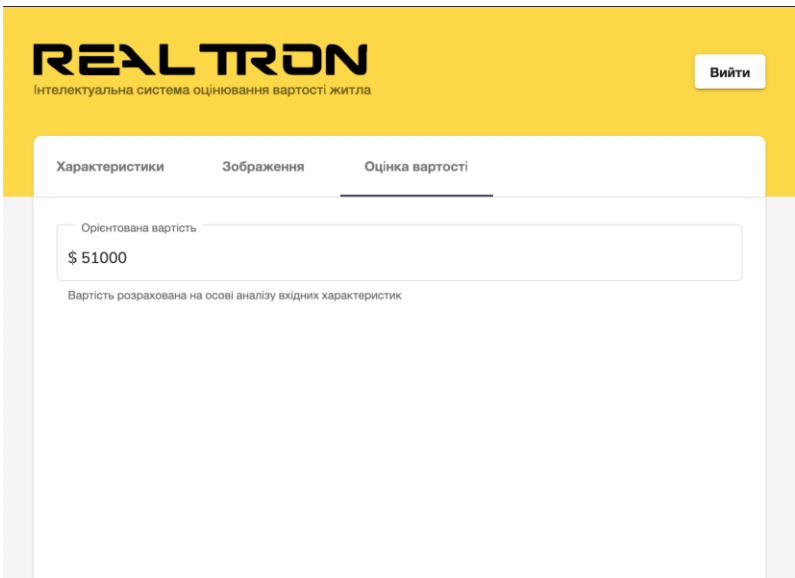
Сторінка авторизації



Сторінка введення характеристик



Сторінка завантаження зображень

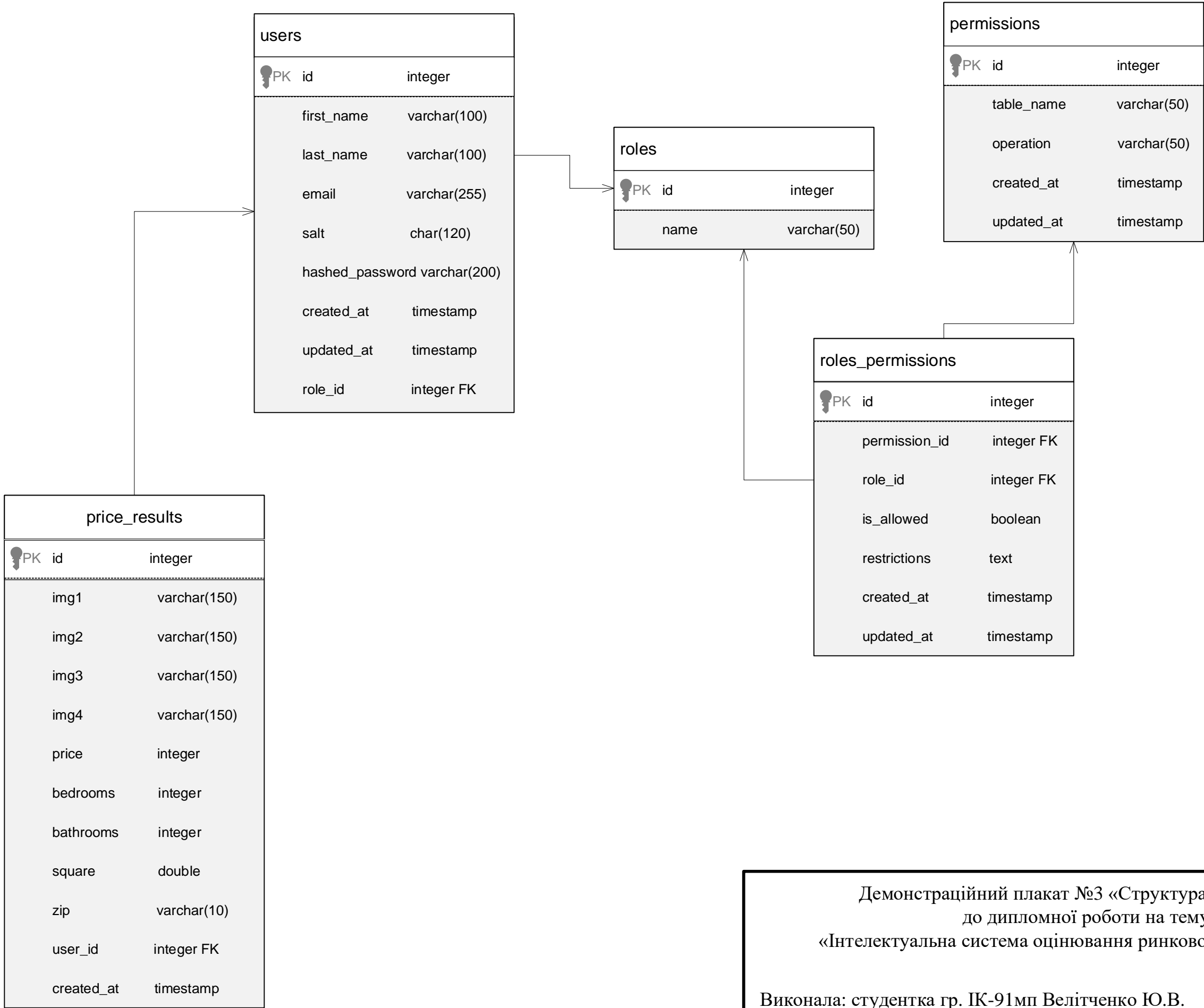


Сторінка виведення результату

Демонстраційний плакат №2 «Інтерфейс користувача»
до дипломної роботи на тему
«Інтелектуальна система оцінювання ринкової вартості житла»

Виконала: студентка гр. ІК-91мп Велітченко Ю.В.
Керівник: д. т. н., доцент Корнага Я.І.

Структура бази даних



Демонстраційний плакат №3 «Структура бази даних»
до дипломної роботи на тему
«Інтелектуальна система оцінювання ринкової вартості житла»

Виконала: студентка гр. ІК-91мп Велітченко Ю.В.
Керівник: д. т. н., доцент Корнага Я.І.

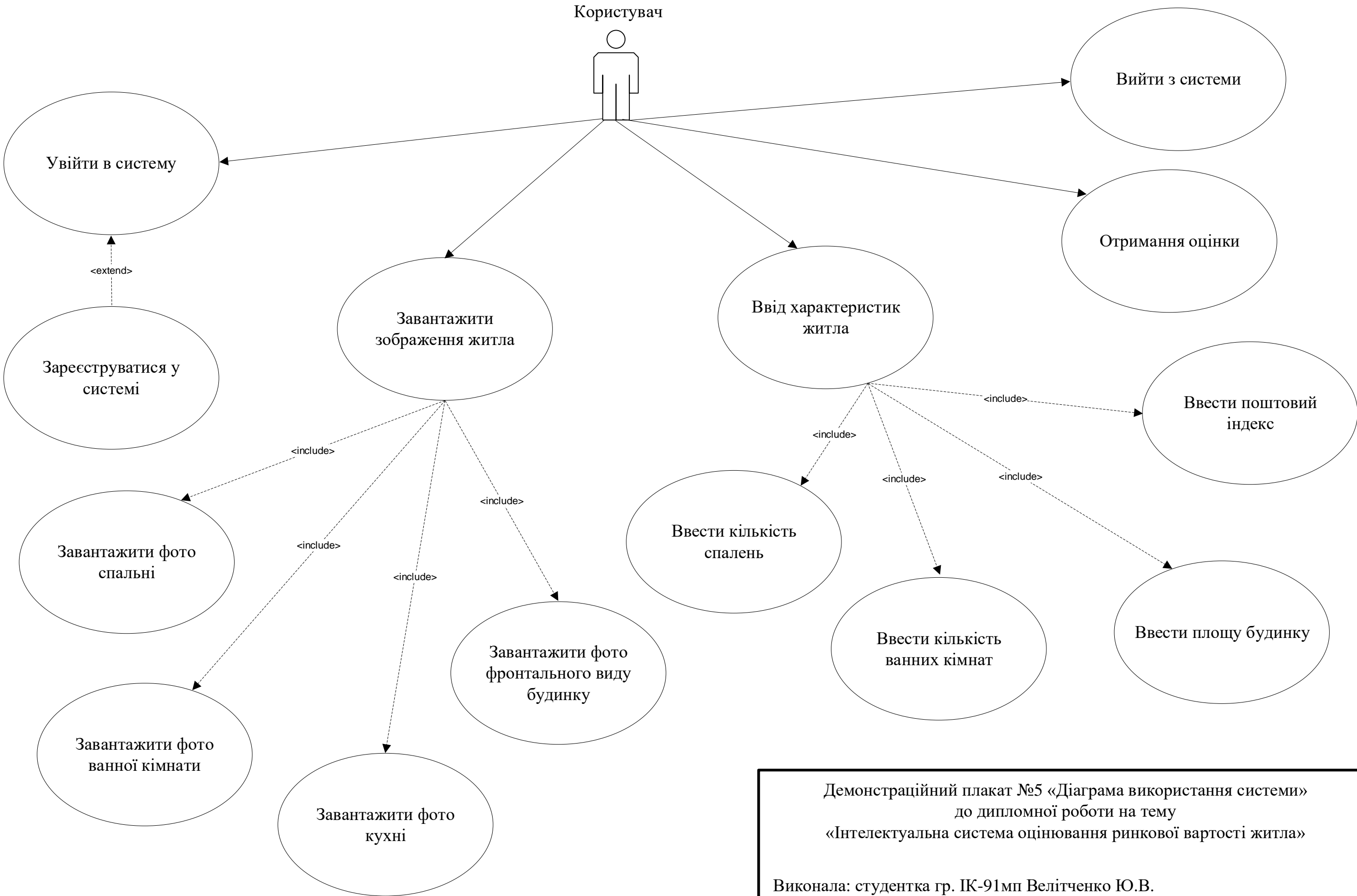
Модель нейронної мережі



Демонстраційний плакат №4 «Модель нейронної мережі»
до дипломної роботи на тему
«Інтелектуальна система оцінювання ринкової вартості житла»

Виконала: студентка гр. ІК-91мп Велітченко Ю.В.
Керівник: д. т. н., доцент Корнага Я.І.

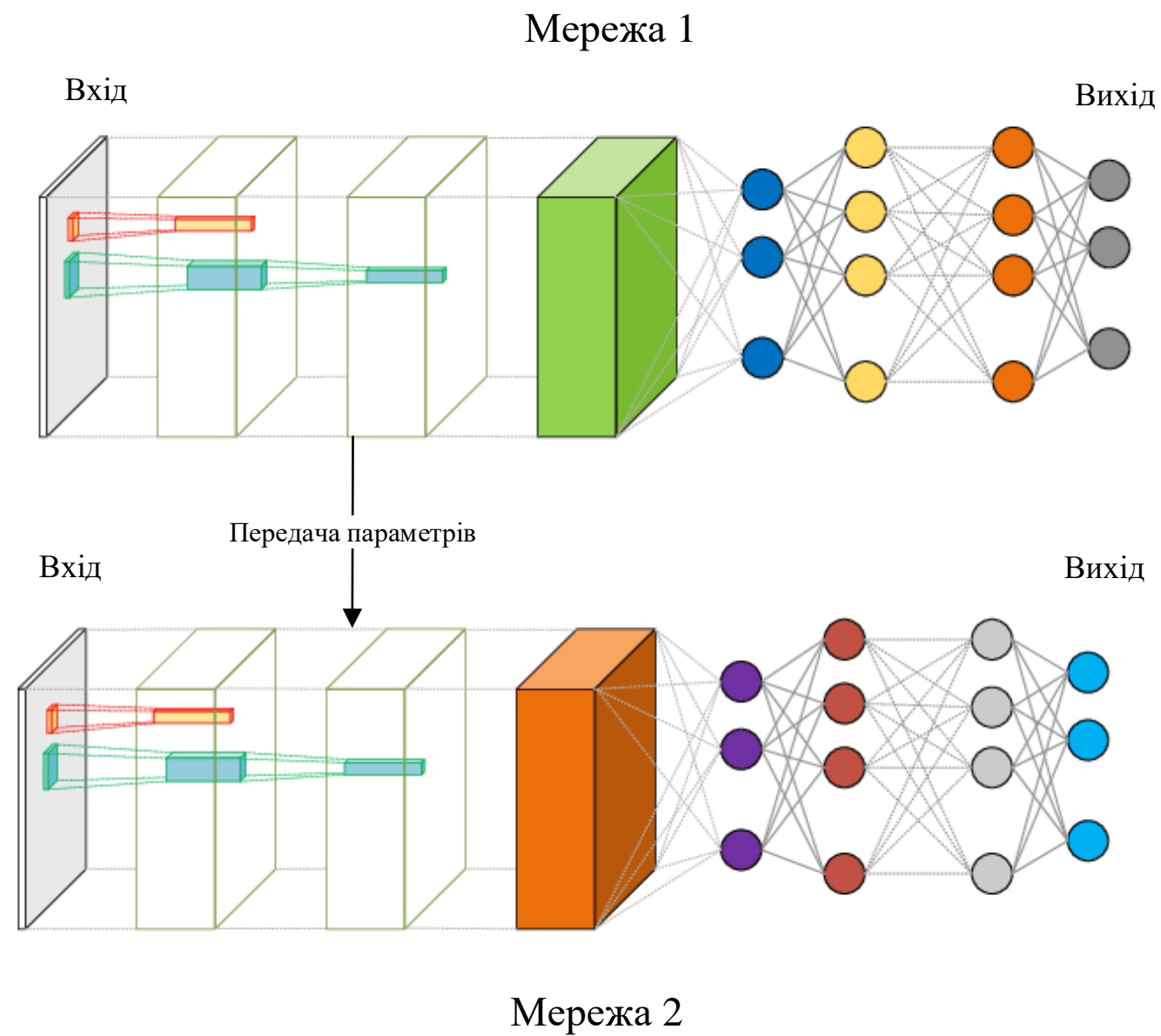
Діаграма використання системи



Демонстраційний плакат №5 «Діаграма використання системи»
до дипломної роботи на тему
«Інтелектуальна система оцінювання ринкової вартості житла»

Виконала: студентка гр. ІК-91мп Велітченко Ю.В.
Керівник: д. т. н., доцент Корнага Я.І.

Трансферне навчання нейромережі



Демонстраційний плакат №6 «Трансферне навчання нейромережі»
до дипломної роботи на тему
«Інтелектуальна система оцінювання ринкової вартості житла»

Виконала: студентка гр. ІК-91мп Велітченко Ю.В.
Керівник: д. т. н., доцент Корнага Я.І.

Ім'я користувача:
Лісовиченко Олег Іванович

ID перевірки:
1005428039

Дата перевірки:
10.12.2020 23:46:05 EET

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
10.12.2020 23:49:39 EET

ID користувача:
76913

Назва документа: Велітченко_IK-91мп

Кількість сторінок: 38 Кількість слів: 8868 Кількість символів: 66848 Розмір файлу: 65.42 KB ID файлу: 1005719592

7.39% Схожість

Найбільша схожість: 2% з джерелом з Бібліотеки (ID файлу: 1000704947)

0.83% Джерела з Інтернету

6

Сторінка 40

6.9% Джерела з Бібліотеки

117

Сторінка 40

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0% Вилучень

Немає вилучених джерел